The University Of Sheffield.

Access To Thesis.

- A bound copy of every thesis which is accepted as worthy for a higher degree, must be deposited in the University of Sheffield Library, where it will be made available for borrowing or consultation in accordance with University Regulations.
- All students registering from 2008–09 onwards are also required to submit an electronic copy of their final, approved thesis. Students who registered prior to 2008–09 may also submit electronically, but this is not required.

Author: Jianing Chen Dept: Automatic Control and Systems Engineering

Thesis Title: Cooperation in Swarms of Robots without Comm Registration No: 100202371

For completion by all students:

Submit in print form only (for deposit in the University Library): ☐

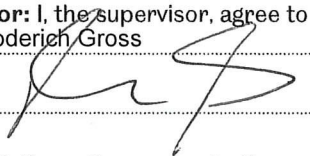
Submit in print form and also upload to the *White Rose eTheses Online* server: In full ☑ Edited eThesis ☐

Please indicate if there are any embargo restrictions on this thesis. Please note that if no boxes are ticked, you will have consented to your thesis being made available without any restrictions.

Embargo details: (complete only if requesting an embargo to either your print and/or eThesis)	Embargo required?		Length of embargo (in years)
Print Thesis	Yes ☐	No ☑	
eThesis	Yes ☑	No ☐	1

Supervisor: I, the supervisor, agree to the named thesis being made available under the conditions specified above.

Name: Roderich Gross Dept: Automatic Control and Systems Engineering

Signed:  Date: 9 March 2015

Student: I, the author, agree to the named thesis being made available under the conditions specified above.

I give permission to the University of Sheffield to reproduce the print thesis in whole or in part in order to supply single copies for the purpose of research or private study for a non-commercial purpose.

I confirm that this thesis is my own work, and where materials owned by a third party have been used copyright clearance has been obtained. I am aware of the University's *Guidance on the Use of Unfair Means* (www.sheffield.ac.uk/lets/design/unfair)

I confirm that all copies of the thesis submitted to the University (including electronic copies on CD/DVD) are identical in content.

Name: Jianing Chen Dept: Automatic Control and Systems Engineering

Signed:  Date: 9 March 2015

For completion by students also submitting an electronic thesis (eThesis):

I, the author, agree that the University of Sheffield's eThesis repository (currently WREO) will make my eThesis available over the internet via an entirely non-exclusive agreement and that, without changing content, WREO may convert my thesis to any medium or format for the purpose of future preservation and accessibility.

I, the author, agree that the metadata relating to the eThesis will normally appear on both the University's eThesis server and the British Library's EThOS service, even if the thesis is subject to an embargo. I agree that a copy of the eThesis may be supplied to the British Library.

I confirm that the upload is identical to the final, examined and awarded version of the thesis as submitted in print to the University for deposit in the Library (unless edited as indicated above).

Name: Jianing Chen Dept: Automatic Control and Systems Engineering

Signed:  Date: 9 March 2015

THIS SHEET MUST BE BOUND IN THE FRONT OF THE PRINTED THESIS BEFORE IT IS SUBMITTED

Cooperation in Swarms of Robots without Communication

Jianing Chen

A thesis presented for the degree of
Doctor of Philosophy

Department of Automatic Control and Systems Engineering

The University of Sheffield

9 Mar 2015

Abstract

Swarm robotics aims to use a large group of relatively simple robots to solve tasks that can hardly be achieved by a single robot in the group. Compared to single robot systems with increased capability, a swarm robotic system may have advantages in robustness, flexibility and scalability. However, designing cooperative behaviors for a swarm robotic system is a challenging problem, especially when the robots may not have communication capabilities and thus only know local information. For a swarm of miniature mobile robots that cannot communicate explicitly, this thesis studies fully decentralized solutions of two problems. For the problem of cooperative transport, the thesis presents a strategy to push an object that is large enough to occlude the robots' perception of the goal of the transportation. For the problem of pattern formation, the thesis investigates algorithms based on the Brazil nut effect that can organize the swarm of robots into an annular formation. These problems are studied using physics-based computer simulations as well as experimental implementations based on the e-puck robotic platform. The simplicity of the solutions make them suitable for applications that require the individual robots to be as simple as possible. Example application scenarios could be micro robot swarms working in the human body.

Declaration

I, Jianing Chen,

declare that this thesis describes an original research carried out by my own. It has not been previously submitted to any university for the award of any degree. Where I have quoted from the work of others, the source is always given. Contents presented in the thesis that were published in the scientific literature are detailed in Section 1.5.

Signature of the author:

Acknowledgments

First and foremost, I want to thank my parents, who always support me unconditionally so that I can fully concentrate on my study.

I couldn't have achieved this thesis without Roderich Groß. As my supervisor, he is a great mentor of doctoral study. He unreservedly passed me his experience in the field as well as chances to advance. Moreover, he never turned away when I need help in scientific literature writing and proof-reading. While being a modest scholar, he is also keen on public engagements that motivate people to embrace science and technology. This made me gain much valuable experience both inside and outside the research work.

The studies in this thesis were conducted in the Natural Robotics Lab, which is a fantastic research group. I want to thank every member of the lab. In particular, I wish to express my gratitude to the following people:

> Melvin Gauci — as the colleague who worked alongside me throughout the four years of my Ph.D study, he helped me countless times with both advices and actions.
>
> Andreas Kolling — whose expertise in mathematics significantly strengthened the theoretical works in this thesis.
>
> Wei Li — more than helping me in numerous experiments, he also shared much knowledge in his research topic with me.
>
> James A. Marshall — who provided some key advices in the studies of social insects behavior.

The Natural Robotics Lab received funding from the 7th European Community Framework Programme via Roderich's Marie Curie European Reintegration Grant (grant no. PERG07-GA-2010-267354).

Contents

Chapter 1

Introduction

Social insects such as ants or bees demonstrate that a large number of simple individuals can accomplish difficult tasks through cooperation. This has inspired the field of swarm robotics, which investigates intelligent behavior that 'emerges' from the interactions of simple robots. Such intelligence (often referred to as 'swarm intelligence' [126]) can let a group of robots solve tasks that may be too difficult for an individual robot to solve.

Instead of pushing the hardware capability of a robot in the system beyond the state-of-the-art, swarm robotics focuses more on making the robots cooperate so that the capability of the system can be extended by increasing the number of robots. Compared to conventional robotic systems, a swarm robotic system offers potential advantages in robustness, flexibility and scalability [105]. Moreover, the benefits of having multiple robots to work cooperatively could also increase super-linearly with group size [22][81]. Theories of self-organization also reveal that randomness or uncertainty in individual behavior may greatly enhance the system's ability to explore new behaviors and find new solutions rather than being harmful [62]. Furthermore, the simplicity of the individual robots makes a swarm robotic system suitable for some application scenarios. Examples of such scenarios could be micro-robot swarms working in the human body, or, dispensable robots in which the cost of each individual needs to be as low as possible.

1.1 Challenges in Swarm Robotics

Communication and Localization

Cooperative behaviors appear in creatures of all levels of intelligence, for example, as advanced as human or as simple as bacteria (e.g. decision-making based on quorum sensing [79]). In robotics, using multiple robots capable of cooperation may reduce the hardware requirement of individual robots in the system. However, robots working cooperatively usually require the capability of communication.

Via communication, the information available for any individual in the group is not limited to what can be acquired by the individual's own perception. For example, in a multi-robot searching task, every robot could know a target's position when one of the robots finds the target and makes a broadcast. Such information that is not necessarily within the reach of a robot's own perception range can be considered as global information.

Technically, sharing global information among a group of robots can be achieved by state-of-the-art technologies, such as WiFi and GPS. However, technologies providing global information are often sophisticated and add limitations to a swarm robotic system [64][108]. For example:

- Addressed communications can introduce a bottleneck in the maximum number of robots working in a same area. For example, connection quality of mobile telecommunication networks can degrade when a large number of devices are used in a small area sharing a same communication channel [144].

- The system may be limited in terms of applicable working environments. For example, radio based communication as a whole can barely work in underwater environments whereas GPS is not suitable for indoor environments.

- The performance or availability of the technology may not scale well with the system. For example, GPS's perception may be sufficient for an outdoor cooperation in urban environments, but it would be insufficient for a swarm of miniature robots on a table.

- The infrastructure may become a single point of failure in the system. For example, a typical WiFi network requires a router, which is a single point of failure.

Thus, a system developed in a research environment using such technologies may have limited applicability to other environments.

Swarm robotics aims at the cooperation of a large number of robots [13][105]. Swarm robotic systems have the potential to operate in a wide range of environments, for example, micro-robot swarms could act inside the human body and under-water searching swarms could act in the deep sea. In these kinds of environments, solutions based on the types of communication technologies listed above may not be valid anymore. While research can be aimed to solve these limitations, to avoid using such types of communication technologies or even global information entirely may be a choice that is more suitable for swarm robotics. Hence, complex communication may not always be helpful in some cases [6].

From Local Information to Global Behavior

A swarm robotic system is expected to accomplish tasks that are meaningful from the global point of view. To make the whole swarm behave in a desired way by designing the behavior of the individual robots is a challenging problem. As we argued in the previous section, the individuals in a swarm robotic system should only rely on local information. Under such a constraint, achieving a desired global behavior becomes even more challenging [73].

In many natural swarm systems, individuals rely only on local information. For example, the perception range of ants and bees are much smaller than the range of their activities. The principles and phenomena behind existing swarms in nature are also important to study in the context of swarm robotics. Often, the relation between the global behavior and the individual behavior is not explicit, and in some cases counter-intuitive [42].

Compared to studying present swarms in nature, designing a swarm robotic system could be an even more challenging problem. Lack of global information can mean that individuals do not know all of the key elements in a task. As a result, conventional ideologies in robotics may not be applied when solving problems in swarm robotics. For example, an absolute position in the environment may not be exchanged among a group of robots without a global reference frame. Thus, any actions based on global positions may not be applied unless the problem of localization is solved first (also known as collective localization [86]). Under such

a condition, tasks as simple as making a group of robots go to a common location (also known as aggregation) may become challenging to achieve.

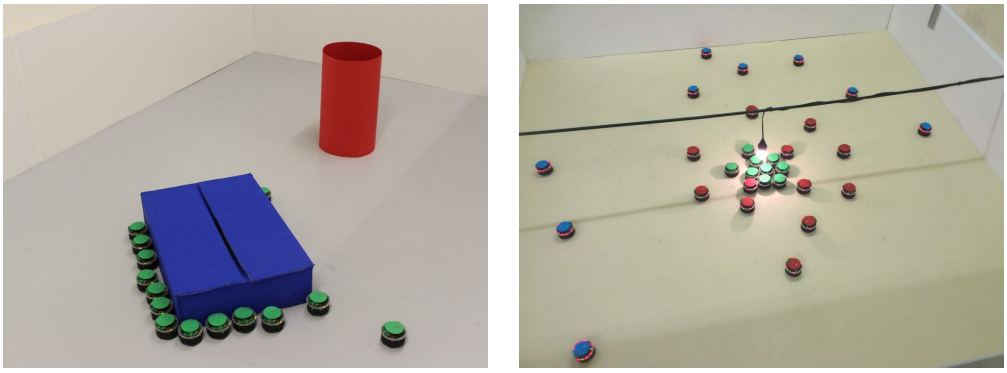
1.2 Problem Statement

This thesis aims to study how a swarm of robots can cooperate without communication. Which kind of interactions can be considered as communication is somewhat subjective.

In general, communication is used to exchange data. For example, a cooperative behavior may be achieved assuming that the robots are capable of broadcasting data packets using infrared range-and-bearing boards. In such a case, the communication mechanism itself (e.g. addressing, modulation and error checking) does not need to be a part of the behavior. Rather, the design of the cooperative behavior employs a method of communication to ensure that the required information is exchanged in a straight forward manner. This kind of communication may be referred to as 'explicit communication' [91].

In some cases, the observation of the presence of an entity in the environment can communicate information and, thus, may be considered as communication. For example, a robot can know whether it is inside a formation by checking the number of neighboring robots. The status of an LED on a robot can also be viewed as a 'Yes or No' message to other robots. In these cases, the mechanisms are typically simple and robust while the process of communication is implemented as a part of the cooperative behavior. These kinds of interactions, if considered as communication, but not explicit (e.g. signaling), are referred to as 'implicit communication' [91]. A particular case where the robots communicate indirectly via applying changes to the environment may also be referred to as 'stigmergic communication' [71].

In our swarm robotic systems, no explicit communication is used. This also implies that localization methods based on explicit communication have to be avoided. To investigate how a swarm robotic system can cooperate without these elements, this thesis investigates two tasks in particular.



(a) (b)

Figure 1.1: The two problems studied in this thesis: (a) a cooperative transport task, (b) a pattern formation task. While the robots are aware of the presence of nearby robots, they are not required to communicate with each other explicitly to solve these task.

Cooperative Transport

In a cooperative transport task, a group of robots need to move an object to a desired position (target position). It is expected that the object needs to be manipulated by multiple robots cooperatively due to its physical properties.

We attempt to solve such a task using a swarm robotic system. Without explicit communication, the target position can hardly be specified for the robots in the form of coordinates in a global reference frame. Therefore, the target position is represented by a physical item (the goal). The goal can be perceived by the robots using their own sensors. In such a scenario, robots manipulating the object typically require consistent perception of the goal. In general, when the object is much larger in size than an individual robot, the robots' perception of the goal may be occluded by the object. Such an occlusion problem usually exists when the robots are pushing the object towards the goal. Fig. 1.1(a) shows such a scenario. The thesis will investigate how the swarm robotic system can collectively overcome the occlusion problem.

Pattern Formation

A typical pattern formation task requires the robots to form a spatial structure that is meaningful from the global perspective. For example, a swarm of robots that are initially randomly located on a plane attempt to form a required shape.

Fig. 1.1(b) shows an annular pattern in which robots of the same color are in a same layer.

Pattern formation can be observed in nature in both passive and active particle swarms. We are specifically interested in the Brazil nut effect [101], where heterogeneous repulsion and homogeneous attraction make a large number of particles form layered structures when being shaken. A swarm of robots may be organized into such a formation when the robots emulate the motion of the particles in such a system. However, to achieve such a motion, a robot may need to know the size of the particle that nearby robots are emulating. Moreover, a consistent perception of a global attraction point may also be needed [51]. We attempt to design and implement an algorithm based on the Brazil nut effect even when these assumptions are violated.

1.3 Contributions

Cooperative pushing of large objects that cause visual occlusion We propose a cooperative transport strategy that can be applied in scenarios where the object is large enough to occlude the robots view of the goal. The individual robots manipulate the object by pushing. They behave only relying on local information acquired from their own sensors and do not need to communicate explicitly with each other. The performance of the group of robots scales well with the number of robots, making it possible to transport objects of various shape and size without adjusting the controller. Through a mathematical analysis in an ideal 2-D environment, the strategy is proved to be able to transport arbitrary convex shaped object to the goal, while a counter example of a concave object is provided. Experiments with 20 robots and objects of different shapes are reported and the results are compared against model predictions.

Human-swarm interaction through an agent robot The goal in the transport strategy can be replaced by a mobile robot. In our experiments, we let a human operator control the 'goal robot' remotely. As a consequence, the group of robots essentially transport the object along the desired direction as defined by the operator. Such an experiment can be viewed as a successful human-swarm interaction in which the human commands a swarm robotic system remotely through

an agent robot.

Spatial segregation without communication We ported onto physical robots an existing algorithm inspired by the Brazil nut effect [51], which makes a swarm of robots form an annular ring formation. The robots emulate the motion of particles of different sizes, and the robots belonging to the same particle-size group will end up being in the same layer in such a formation. The algorithm does not require robots to discriminate among robots of different groups; a robot only needs to know which group itself is in. We test the effects of different particle size ratios and present experiments with up to 30 robots. The different groups were able to segregate with a high level of accuracy.

Decentralized construction of global gradient From the segregation algorithm, we abstracted a fully decentralized method to form a global gradient without explicit communication between robots. Initially, all robots in the swarm have homogenous (or valueless) status. With the algorithm being applied, a radial gradient can be constructed among them. This gradient makes it possible to implicitly decide a center or origin within the swarm.

An e-puck library alongside the official one In this thesis, the robotic platform used for physical experiments is the e-puck [28]. It comes with the official e-puck library. Partially based on this library, we have developed a new e-puck library that overcomes some of the drawbacks of the official e-puck library. The e-puck controllers in this thesis are all programmed based on the new library. The library has been made open to the public under the MIT license on GitHub [24]. Apart from some technical advantages, we believe this library is more user friendly as it has a better interfacing model.

1.4 Thesis Outline

The thesis is organized as follows.

Chapter 2 first provides a background on swarm robotics. Then related works in the problems of cooperative transport and pattern formation are introduced and discussed.

Chapter 3 focuses on the technological developments of this thesis, which provided the framework for the case studies.

Chapter 4 presents the occlusion based cooperative transport strategy, including a detailed controller description, two series of systematic physical experiments and two computer simulations.

Chapter 5 describes the physical implementation and experiments about the Brazil nut effect inspired segregation algorithm, which was presented in previous research [51].

Chapter 6 introduces an improved algorithm to achieve segregation which overcomes some of the requirements of the algorithm presented in Chapter 5.

Chapter 7 concludes this thesis and discusses potential directions for future work.

1.5 Related Publications

Some of the contents presented in the thesis were published in the scientific literature. In the following, these contents are summarized.

A preliminary version of the simulation study in Section 4.7 was published in:

- J. Chen and R. Groß. Cooperative multi-robot box pushing inspired by human behaviour. In *Proceedings of the 12th Annual Conference on Towards Autonomous Robotic Systems (TAROS 2011)*, volume 6856 of *Lecture Notes in Computer Science*, pages 380–381, Berlin, Heidelberg, 2011. Springer-Verlag.

A preliminary version of the implementation in Section 4.4 and the technical development introduced in Section 3.1 were published in:

- J. Chen, M. Gauci, and R. Groß. A Strategy for transporting tall objects with a swarm of miniature mobile robots. In *2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, pages 863–869. IEEE Computer Society Press, Los Alamitos, CA, 2013.

The experiments and the mathematical analysis in Chapter 4 were published in:

- J. Chen, M. Gauci, W. Li, A. Kolling and R. Groß. Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots. In *IEEE Transactions on Robotics*, in press.

The implementation and the experiments using two groups of robots in Chapter 5 were published in:

- J. Chen, M. Gauci, M. J. Price, and R. Groß. Segregation in swarms of e-puck robots based on the Brazil nut effect. In *Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2012)*, pages 163–170, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

In addition, some of the technical development in Section 3.1 had also contributed to the work reported in the following publications:

- M. Gauci, J. Chen, T. J. Dodd, and R. Groß. Evolving aggregation behaviors in multi-robot systems with binary sensors. In *Proceedings of the 2012 International Symposium on Distributed Autonomous Robotic Systems (DARS 2012)*, volume 104 of *Springer Tracts in Advanced Robotics*, pages 355–367. Springer-Verlag, Berlin, Germany, 2014.

- M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Self-organized aggregation without computation. *The International Journal of Robotic Research*, 33(8):1145–1161, 2014.

- M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Clustering objects with robots that do not compute. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2014)*, pages 421–428, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

Chapter 2

Background and Related Works

2.1 Background

The field of cooperative mobile robotics[1] investigates the use of multiple robots to solve tasks cooperatively. It has been established for decades [2]. Cooperative task solving behaviors are widely observed in animal kingdom. In the robotics context, such behaviors are of interest for a number of reasons. For example, the reliability of a single robot tends to decrease as its capability and thus complexity is increased. Therefore, making a number of simple robots that work cooperatively may be more suitable in some circumstances, in particular where the overall system remains intact despite failure in some of its robots. Moreover, some types of tasks are inherently suited for a multi-robot system [22]. Tasks involving searching are examples of these.

A swarm robotic system uses a large number of simple robots to solve tasks cooperatively. It can be categorized under multi-robot systems but the differentiation from a multi-robot system to a swarm robotic system could be more substantial than just the number of robots [105]. For example, swarm robotic systems tend to be a group of decentralized and homogeneous robots. The emphasis on the simplicity of an individual robot in a swarm robotic system could be stronger than in a multi-robot system. Swarm robotics is a domain focused on studying methodologies, phenomena and applications of swarm robotic systems. This section will give a brief survey around the common topics in the domain of swarm robotics.

[1] also referred to as distributed autonomous mobile robotics

2.1.1 Origins and Terminology

"Swarm intelligence" is a rather inclusive term to describe the essence of swarm robotics. It originates from biological research around swarms in nature, especially social insects [41]. It can be used to describe the high level intelligence of a swarm that surpasses the intelligence of its individuals. The ant colony is an iconic example of such a swarm. To analyze the swarm as an intelligent system to solve a task, the intelligence required to overcome the difficulties in the task is not necessarily the intelligence of an individual in the swarm. For example, a single ant may not have enough brain power to work out the shortest path to a prey while the shortest path can be achieved through a chain of causes and effects. Though behaviors to form a shortest path are still based on intelligent choices made by each ant, the intelligence behind these choices seem to be much more primitive than the final outcome it achieves. Such intelligence may be viewed as swarm intelligence.

Though swarm intelligence is generally a wider area than swarm robotics, the concept of swarm robotics probably formed first. In 1988, a paper titled "The Concept of Cellular Robotic System" [12] was published. The cellular robotic system proposed in the paper distinguished itself from conventional distributed robotic systems as a robot in the system tends to communicate with only nearby robots (typically direct neighbors). In [14], the relation between the cellular robotic system and swarm intelligence has been discussed while the term "robot swarm" was used to describe a type of cellular robotic system. In [35], a taxonomy of *swarm robots* was provided. In [13], the term "swarm robotics" appeared to be formed as well as many definitions and motivations about it were discussed.

In [105], Sahin proposed a definition of swarm robotics: "Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment." He also proposed explicit criteria for distinguishing swarm robotics research from other research. It was also emphasized that these criteria are subjective and are not intended to impose restrictions to studies. Nevertheless, a multi-robot system following these criteria may inherit some of the advantages of being a swarm robotic system. The criteria are summarized below with their possible advantages analyzed:

1. **Autonomous robots.** Each of the robots needs to be autonomous and interact with the world physically. This sets swarm robotics apart from intelligent systems without mobility such as virtual agent-based systems or distributed control systems [85].

2. **Large number of robots.** In [105], it was suggested that a swarm robotic system should *be able to* have more than 10 robots. In research and experiments, it is not always necessary to use a large number of robots. However, the design should make it possible to directly cope with a large number of robots. For instance, a system studied with 8 robots could be expanded to 15 to 20 robots without changing the design. This specification is strongly related to the advantage of scalability. To guarantee scalability, the number of robots in a design should also be dynamic. If instead, a design only works with a fixed number of robots (e.g. 20), building a single robot with the total capability of such a group may be more efficient where the task scenario allows.

3. **Few homogenous groups of robots.** Ideally, all of the robots in a swarm robotic system share the same design, both in their controllers and hardware platforms. Differences between the individual robots in the swarm can however be accepted. Where the robots sharing the same design define a group, it should be avoided that the total number of groups is significant when compared to the total number of robots. By following this specification, the robustness of swarm robotic systems may be inherited. Otherwise, the failure of a critical group may cause the whole swarm to fail.

4. **Relatively incapable or inefficient robots.** The incapability of an individual robot is relative to the task. A swarm robotic system should be able to solve tasks that a single robot in the swarm cannot solve efficiently. In some cases, the use of a swarm of robots makes a difference to the feasibility of the task, such as in cooperative searching tasks or cooperative transport tasks. In some other cases, cooperation between individuals can increase the capability of the individual super-linearly. For example, the pheromone trail of ants can optimize the route of foraging [19]. One advantage of studying swarm robotics using on simple robots in a research environment is that:

the solution may inherently rely less on sophisticated technologies that often have restrictions in applicable environment and poor scalability.

5. **Local sensing and communication capabilities.** The perception of the environment and the communication between the robots need to be achieved by the robot's own sensors. In [105], it was also suggested that the robot can only communicate with a very limited number of robots near it. This point has also been highlighted in Chapter 1 because it is related to most of the advantages of swarm robotic systems. For example, the robustness of a cooperative behavior may be enhanced if it is achieved with minimum communication [4]. Many communication technologies are centralized systems by themselves, which introduce a single point of failure to any swarm robotic system utilizing them. The capacity of a communication network can introduce a bottleneck to the scalability. The communication hardware may also not be scalable in the aspects of physical size and applicable environment.

2.1.2 Physical Entities

Because swarm robotics does not focus on solving problems by extending the capability of individual robots beyond the state-of-art, the hardware of the robots used in research can be compromised when the scope of research needs to be limited. However, low individual capability may not be a necessary condition of a swarm robotic system. A swarm robotic system in future could consist of a large number of humanoid robots built with cutting-edge technologies. Furthermore, low individual capability generally does not mean the basic mechanism of the robot needs to be common; novel but simple mechanisms may also be pursued in swarm robotics. Therefore, hardware development is still an important topic in swarm robotics. This sections will introduce a number of present hardware platforms or concepts related to swarm robotics.

Mobile Vehicles

A swarm robotic system can be a large number of conventional mobile vehicles, for example, differential wheeled robots, legged robots or unmanned aerial vehicles. When the number of robots are large, the cost to build and operate a large number

of robots will be high. Therefore, suitable robot platforms may be needed to reduce such costs, especially for research purposes. Such platforms primarily aim to serve as research and education platforms. They are typically used indoors, for example, on desktops. Depending on the research topic, they may only be used in experiments as abstract models of the agents in the system to be studied.

Many research and education platforms that are suitable for swarm robotic research were presented, for example, Alice [23], Jasmine[2], Khepera[3], etc. In the following, a selection of these platforms are discussed in more detail.

1. e-puck[4] [28] [82]. It is a differential wheeled robot with a cylindrical body of diameter 7.0 cm. It has a series of simple sensors: a directional camera, eight proximity sensors, a three-axis accelerometer and three microphones. The proximity sensors are distributed around its body to collect local information within an approximate range of 8.0 cm. The directional camera is facing to the front. It is only able to capture pictures with very low resolution (e.g. 40×40). Because the hardware mechanism of e-puck is very generic, it is used as the implementation platform in most of the studies in this thesis.

2. Swarm-Bot (s-bot)[5] [84]. It has cylindrical body with a diameter of 12.0 cm and a height of 19.0 cm. It is a differential wheeled robot with additional tracks (sometimes referred as "treels"). The design enables the S-bot to move in outdoor environments with a moderately rough ground. It has a series of sensors, including an omni-directional camera. Moreover, it has a gripper. This gripper can be used to form physical connections between the robots. Thus, the s-bot can be used to study self-assembling robots. For example, a group of s-bots can assemble themselves into a larger robot to overcome gaps in the ground [46].

3. Kilo-bot[6] [102]. Its diameter is 2.8 cm. It is driven by two paper motors, which use vibrations to cause the robot move slowly. The only sensor it has is an infrared range-and-bearing sensor facing the ground. Via the reflection

[2] http://www.swarmrobot.org/
[3] http://www.k-team.com/mobile-robotics-products/khepera-iii
[4] http://www.e-puck.org/
[5] http://www.swarm-bots.org/
[6] http://www.eecs.harvard.edu/ssr/projects/progSA/kilobot.html

and diffusion of the floor, this sensor can broadcast very simple messages to nearby kilo-bots within a range around 9 cm. It was designed to be low-cost; each kilo-bot is claimed to be made with only $14 worth of parts. The systematic design of the kilo-bot platform also makes it possible to program, command and charge large number of kilo-bots simultaneously. These features enable physical experiments with hundreds or thousands of robots. For example, over one thousand kilo-bots were used in [104], which presents an algorithm to organize a thousand-robot swarm into any desired pattern.

Modular Reconfigurable Robots

Cooperation between robots can be enhanced by new types of robot hardware that exploit physical interactions between the robots. A modular reconfigurable robotic system is a group of typically homogeneous robots that are capable of making physical connections with each other [151]. Once connected, the group of robots may act as one robot with increased capabilities. The group of robots may also assume different configurations to adapt to different tasks. Compared to a single robot with comparable capabilities, a modular reconfigurable robotic system has advantages in versatility, robustness and low cost [150].

In a modular reconfigurable robotic system, the capability of a single module may be reduced to 'atomic' level, for example, one module has only one degree-of-freedom (DOF) in terms of interacting with the environment. A representative design can be found in [153] which was later developed into the (manually reconfigurable) "Molecubes" [152]. The overall shape of body is a cuboid that is composed of two tetrahedrons. On each of the six faces of the cuboid, connection mechanism are presents. The only actuator in the module is a servo motor that changes the relative angle of the two tetrahedrons. In other words, each of the module can act as a joint with one DOF. However, a number of modules can be combined in to a super structure that may be as flexible as a human armature.

Micro Robotics

Less individual capability implies that the individual robots can be made very simple in hardware. Thus, swarm robotics and micro robotics may become more

and more related in future.

In [112], an introduction of micro or nano robotics was given and challenges in such a domain were outlined. The most obvious difference between the micro robotics and conventional robotics is the hardware design caused by the requirement of size. Not all of the present hardware technologies can scale with the size of the robot to micro or nano meter level. Therefore, actuation, sensing, power source and manufacturing methods need to be researched. Nevertheless, the hardware capability of a single robot in a micro robotic system will always be extremely limited. The differences are not only in the robot platform. The working environment of a micro robotic system may also be vastly different from conventional robotic systems. For example, "due to surface to volume ratio increases inversely proportional to the scaling factor, therefore, surface properties and forces start to dominate bulk properties and forces." [111]

2.1.3 Common Methodologies

Inspiration From Nature

In [41], swarm intelligence is described to be "deeply embedded in the biological study of self-organized behaviors in social insects". The primarily methodology of studying swarm robotics is to take inspiration from nature, for example, modeling and simulating behaviors of social insects [17].

Social behavior of ants is a frequent subject of research in swarm robotics due to the contrast between the simplicity of their individuals and the complexity of the organization [64].

In [65] and [63], a model of cooperative transport in ants was studied and several robots controlled using a finite state machine successfully transported a large object towards a location indicated by a light spot. The transition between task states or to some specific behaviors were made from perceptual cues. In biology, perceptual cues are the narrow range of stimuli and situations that an animal uses to solve perceptual tasks [145]. In robotics, it is a series of highly abstracted orthogonal states computed from the sensors.

In [32], the ant society was described as a self-organized super-organism while the individual behaviors might be as simple as molecules in a chemical-physics system.

Sometimes, research about swarm systems in robotics and biology make cross-contributions to each other. Some interesting high-level behaviors in biological systems may be hard to be understood when being studied using conventional methods. Using swarm robotics to replicate the same behaviors can provide a strong support in such studies.

For example, in [56], the behavior model of individual fish in a fish school was studied by modeling and simulating the motion of individual fish to achieve a same schooling behavior. Assuming that an individual fish takes only the position and orientation of its nearest neighbor as external perception, a hypothesis about the behavior model of the individual fish in a real fish school was proposed. It was shown using computer simulations that the collective motion of a school of fish using the behavior model achieved similar collective motions as a school of real fish. [100] analyzed the case where some of the simulated fish in a fish school execute an attraction/repulsion function that is different to the one executed by the other fish of the school. Simulations showed that significant changes will appear in the moving trajectory of the school.

Some researchers have also investigated hybrid societies composed of both robots and animal collectives, for example, see [53][132].

Because taking inspiration from nature could possibly be the fundamental methodology of swarm robotics, many related works introduced in this chapter also applied it either explicitly or implicitly.

Evolutionary Approach

A critical problem in swarm robotic systems is that the cause-and-effect between global behavior/phenomena and individual behaviors is not intuitive. Given the desired global behavior, it is difficult to work out what the individuals need to do. One method to address this problem is the evolutionary approach.

To apply the evolutionary approach in a homogeneous swarm robotic system, the individual robots in the swarm are often given a relatively universal controller (e.g. neural networks [38]). By mutating/crossing over the parameters in such controllers, large variety of behaviors can be obtained. Typically, the fitness of a generation in the simulation is evaluated based on the global behavior. In other words, evolution is applied on the individuals while the fitness depends on the

performance of the whole swarm.

In [110], Karl Sims presented a work centered around evolving embodied agents ("virtual creatures") in computer simulations. In addition to the behaviors, the work also evolved the morphologies of the agents. In [109], he presented an artificial evolution in which the creatures compete directly with each other in the virtual world.

In [128], a evolved controller that can aggregate a swarm of robots in computer simulation was published. In [33], an extended series of studies about evolving self-organized behaviors using s-bots was presented. The artificial evolution used a physics-based simulation framework. Multiple global behaviors have been studied, including aggregation, coordinated motion, group transport, etc. The evolved controllers were claimed to be "simple but effective". Analysis showed that the controllers also reserved the advantages in scalability and robustness as a swarm robotic system. In further works, some of the controllers of the individual robots were also tested on physical s-bots. Experiments showed that the controllers obtained in computer simulations also worked on the real robots [46][129].

When the controller synthesis is handed over to computers, the quality of the controllers may be controlled. This gives the possibility to vary other aspects in a system, for example, challenging the convention that a swarm robotic system should be homogenous. In [135], the possibility of a swarm robotic system to be heterogeneous was studied. The swarm were divided into different groups; the controllers for the robots within each group were identical. Multiple strategies for evolving such a swarm were studied. One representative method was that the fitness evaluation and genetic composition were done group-wise. However, the experiments did not show evidence of the effectiveness of such method. The general conclusion of this paper was that homogenous systems were more suitable for the evolutionary approach.

Machine learning is another relevant process of automated controller design in swarm robotics. The review paper presented by Panait and Luke [90] summarizes such methodologies.

2.2 Collective Foraging

2.2.1 In Nature

Swarm intelligence was born from the study of biology in social insects [41]. In many social insects, the capability of a swarm differs from the capability of an individual significantly. Typically, the capability of an individual is insufficient to support over long durations. Thus, for many social insect swarms, self-organization is critical and "deeply embedded in their biology" [41] such that a swarm may be treated as one functional unit.

For example, the self-organization of bees is based on several simple rules in the behavior of individuals. With a large number of individuals, the variation in the behavior of an individual makes the bee colony very flexible and capable of carrying out multiple tasks in different conditions [21].

One of the most fascinating behaviors present in social insects is foraging. Foraging is a systematic task, which includes a series of behaviors like searching, hunting, transporting, etc. Moreover, it often requires advanced interactions with the complex world, for example, localization and navigation. Yet, it can be achieved by simple individuals in some social insects. In some cases, it can be achieved without explicit communication between the individuals [22]. Therefore, foraging behaviors have been studied both in social insects and cooperative robotics. This section will review some of these studies.

Foraging of Ants

Foraging of ants is probably the most well known collective foraging behavior in social insects. Ants could transport their prey back to the nest following an 'invisible highway' across the field. This highway is a trail of pheromone which is deposited by many species of ants [19]. And it is this pheromone trail that is the key mechanism to organize the foraging of ants.

In [54], the functionality of the pheromone trail and the mechanism of recruitment in ants foraging have been described. When a prey is discovered by an ant, it could go back to the nest following the pheromone path left behind. Then, in the nest, it will use multiple (e.g. head waggling and tactile stimuli) ways to recruit members to use this path. Each of the ants recruited will reinforce the

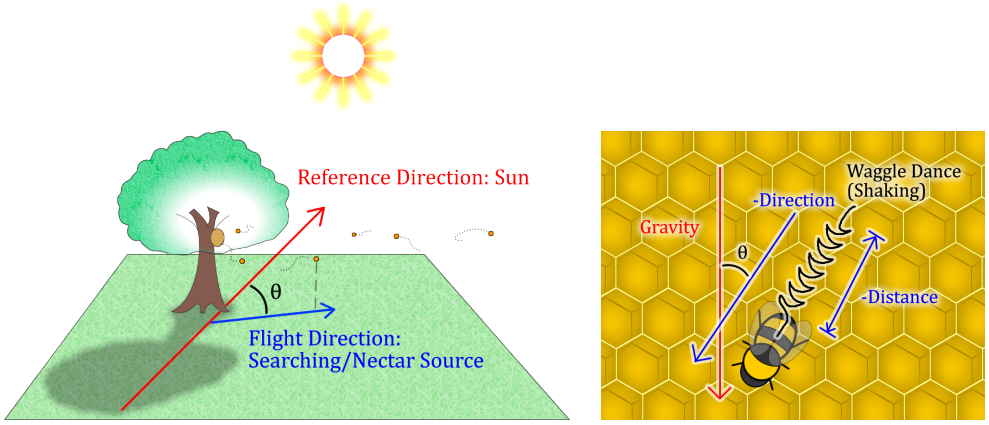


Figure 2.1: (a) How bees know their flight direction on a horizontal plane using the sun as a reference direction. (b) How direction and distance information are communicated using the waggle dance in the hive (vertical planes) using gravity as a reference direction.

pheromone trail which may cause the pheromone density on this path to become stronger than on other paths. Thus, more ants in the nest will be attracted to this path. With the composition of these behaviors, a positive feedback (snowball effect) on the number of transporting ants between the food source and the nest is formed, therefore the growth curve at the beginning of this recruitment is logistic. After some time, the labor in the ant colony will focus on this transporting route. Because a pheromone trail will become weaker and weaker if it is not reinforced by the ants due to saturation of the transportation group size, the number of ants in this transporting path will begin to converge to a peak and then gradually fall to null.

The property of the pheromone trail of ants also provides optimized solutions to the transport route [11]. When an ant meets a branch, it will first choose a random side to travel. Since the longer route will have less average coverage rate of ants than the shorter route, the pheromone strength on the two routes will become different. When the difference in strength is significant enough, the ants will choose the branch with higher pheromone strength which is the shorter route.

Nectar Collecting of Bees

The foraging of a hive of bees is mainly targeted at the nectar from flowers in a wide area around the beehive. Two relatively advanced elements are found to play an important role in the nectar collecting of bees: sun guidance and the waggle dance [97]. The sun provides a global reference direction across the working region of a bee colony, so all bees are essentially equipped with a 'biological compass'. Combined with the odometry of distance which is approximately proportional to the flight time, a bee knows its relative position to the hive. After a bee finds a nectar source and returns, it passes the approximate position of the nectar source to others workers in the nest through a special movement known as waggle dance.

The waggle dance involves running through a small pattern similar to a figure-eight: the bee will first run straight for a short distance with the wings vibrating at approximately 260 Hz. The distance of this run is proportional to the distance between the hive and the nectar source. The relative angle between the running direction and the vertical down (direction of gravity) is same as the bearing of the nectar source with respect to the hive when using the bearing of the sun as the reference [20]. An illustration of how direction and distance are expressed in the waggle dance is given in Fig. 2.1. Similar to recruitment in ant colonies, the recruited bees will also recruit other bees which introduce a positive feedback in the number of engaged workers. Thus, while the working region of a bee colony could be several square kilometers, most foragers in the colony will work effectively in transporting the nectar back to the hive from the best nectar source.

2.2.2 Multi-robot Systems

In swarms of ants and bees, the mechanisms to localize and communicate the position of the prey may be the two critical elements to solve the task of foraging. The mechanism of pheromone trail in ants colonies attracted research interest because it has potential to be implemented in complex environments. However, to replicate a pheromone laying and sensing mechanism using chemical based solutions as used by ants is difficult for present technology; most research attempt to emulate the key characteristics of a pheromone trail.

In [133], a "trail laying and following algorithm" inspired from ants was used to organize a group of robots to search and retrieve resource in an unknown environ-

ment. When an individual robot is searching a resource location, it will memorize its path using odometric localization. Once it finds a resource location, the path to the location — the trail — can be shared among the robots via a communication network. An experiment using a group of four robots in the corridor of an office building was successful. The results of the experiment also showed that sharing a global location by sharing the path to it (rather than its coordinates alone) might help the system to overcome the problem of accumulated errors in odometric localization.

An emulated pheromone trails can be called virtual pheromone trail. In [123], based on a infrastructure called "Virtual Dynamic Environment for Autonomous Robots (V-DEAR)", pheromone trails of ants were emulated. V-DEAR includes a projector that can project images on to the floor of a robot arena. The robot was equipped with a color light sensor in the front that could detect the projected virtual pheromone and a passive infrared sensor in the bottom to recognize the nest area. The behavior of an individual robot was similar to an ant. The system had been tested in both computer simulation with dozens of robots and in real world with three robots. The results showed that the virtual pheromone trail was still very effective in the foraging task. Furthermore, in the simulation, it was verified that the characteristics of pheromone may directly affect the performance of the foraging task. For example, the lower evaporation rate of virtual pheromone would generally lead to higher efficiency (collected food per unit time) in the test environment.

In [93], a virtual pheromone trail was realized using a decentralized mechanism. Each robot in the system is equipped with an infrared range-and-bearing board which consisted of eight infrared transceivers around its body. Each robot will constantly broadcast a specific data packet that contains a type field, a hop-count field, and a data field. The type field distinguishes what type of the virtual pheromone the signal represents. The hop-count field essentially counts the number of times a data packet was relayed. After a robot receive a data packet, it will decrease the hop-count and then relay it. When a robot receives the same type of pheromone from multiple directions, the one with the highest hop-count value is used. This mechanism can create a pheromone gradient which is able to provide multiple types of information about a node within the entire virtual pheromone network.

Another behavior similar to the virtual pheromone trail is self-organized formation. In [89], an experimental study of control policies that let a group of robots forage by forming a chains was presented. In the foraging task, the robots would work in three different roles through automatic division of labor. One of the group will form a chain while explore the environment for the prey. After the prey was found, some robots would be recruited to the front of the path through the path formed by the explorers. Then, those recruited robots would become a retriever which would transport the prey back to the nest along the formed path. When the prey on a path was being transported back to the nest, the path formation would decompose and the robots which formed the path would join the transporting. Therefore, the chain formation established was a path between the nest and the prey which guide all robots to work between the two locations. The robots were equipped with an omnidirectional camera and a color LED ring, which enabled them to communicate simple messages through vision. For example, the robots could determine the direction of the path formation via the color sequence of the robots forming the path.

[127] presented a navigation algorithm for a swarm of robots to perform a dirt cleaning task. The task is similar to a foraging task in that the robots need to retrieve objects from two locations far from the nest. The navigation algorithm is based on a signal propagation mechanism inspired from slime mold. Similar to pheromone trails of ants, this navigation algorithm also enables the robots to form the shortest path between two locations.

2.3 Cooperative Manipulation

2.3.1 In Nature

While a swarm of ants is capable of foraging across a large field based on pheromone trails, cooperation also exists at local level. Large prey that can not be moved by a single ant may be moved cooperatively by a small group of ants.

The cooperative manipulation of a prey may begin with an uncoordinated period in which ants try to handle the prey from any direction [117]. The conclusion about this disordered phase was that it helps to rotate the prey to an orientation in which lower friction force will incur. In [67], the transition into an organized

phase was characterized by an increase in the linear speed of the load and a decrease in the sum of the ant interaction forces. In other words, the coordination of forces will form gradually after the prey starts moving. The embodiment of the prey can be viewed as a media in this process. Through touching the prey, an ant essentially knows how other ants are attempting to move the prey.

Focusing on a species of ants, Pheidole oxyops, [30] tested the hypothesis that a manipulation group may re-orient food items of various shapes to reduce drag. It was found that turning of a item often involves a single 'steering' ant within the group. Any ants in the group can take the place of the steering ant if there are no steering ants. The paper also implied that the turning behavior is a property that emerges from the combination of pre-existing retrieval behavior and the underlying physics of large items.

In [31], a type of more sophisticated cooperative manipulation in army ants was described. To lift a large item, "one worker, usually of larger size, straddles an item at the front while one or more smaller workers help to lift at the back". According to the observations, the foraging task can be improved by the cooperative manipulation of large items. For example, moving the entire prey directly can save the time spent on splitting the prey.

In [15], cooperative manipulation of a large prey was studied among a group of ants, Aphaenogaster Cockerelli. In this study, a small device which has several high sensitivity springs around it and high visibility spots on its center part and the end of those springs was used as a dummy prey. Therefore, when this device was being moved by several ants, its position as well as strength and direction of the force applied by each ants could be tracked by a video camera installed above the transporting site. According to the experimental result, the ants will pull the prey if they are in front of the prey and push the prey if they are behind the prey, which means they knew the direction of transporting from the pheromone trail.

2.3.2 Multi-robot Systems

Over the past 20 years, multi-robot object transportation has become a canonical task for studying cooperation in a group of robots. The three most common types of strategies are pulling, pushing and caging.

Prehensile Systems

In a prehensile system, each of the robots in the group has manipulator(s) that can grasp or hold the object. When a robot is grasping or holding the object, it physically connects itself to the object and can assert forces freely in terms of the direction. To cooperatively transport the object to a goal, the robots can connect to the object and simply move towards the goal in a coordinated way.

In nature, transport through grasping seems to require relatively little coordination of the individuals [15]. Many studies have attempted developing grasping based cooperative transport systems [143][136][58][27], but most of them used only a small number of robots with limited scalability.

In [83], a series of demonstrations involving using a swarm of s-bots [88][34] was presented. The robot can use its gripper to grasp the object. The hardware design of s-bots also enables a robot to grasp the hull of another s-bot. In the task of cooperative transport, some of the robots may choose to pull those robots that were already connected to the object. As a result, several chain formations could be formed, each of them connected to the object. Such a formation made the pulling forces contributed by each of the robots stack up in an efficient way. Therefore, the performance of the system could scale well with group size [47].

A decentralized pulling algorithm for a group of s-bots was published in [52]. In the experiment, the transport group was composed of up to 6 robots. Only some of the robots (typically one) in the groups was able to perceive the goal directly. They could push/pull the object using a simple algorithm. The other robots (the blind robots) could not perceive the goal. A blind robot could estimate the speed and orientation using an evolved neural network [48]. The neural network took only the inputs from the robot's own force sensors. The force sensors were used to measure the force received from the object with respect to the chassis of the robot. The experiment showed that the algorithm was able to make the group perform almost as efficient as if all robots were able to perceive the goal.

The grasp manipulation can only be applied to objects suited for grasping. One alternative way to cooperatively moving an object is to place the object over the top of a number of robots [116]. In order to keep the object above the robots while the transport group is moving, the speeds of the robots need to be planned precisely. In [80], the problem of how to coordinate the motion of

the robots were studied. In order to simplify the motion planning of individual robots, a conceptual robot hand was designed for holding the object. The passive mechanical components in the hand could deal with some complex dynamics in the relative motion between the robot and the object. The robots were controlled by a layered motion planning architecture, which is composed of both centralized and decentralized planners. Within the group, some of the robots were "handling robots". They were firmly holding to the object and transporting the object. The rest of the robots were "regrasping robots". These were the robots that need to adjust their relative positions to the object to achieve a optimal handling.

The object can also be held at its edges by multiple robots around it. A decentralized approach to achieve such a coordination was discussed in [120][121]. The proposed control algorithms and framework were designed for coordinating multiple autonomous mobile robots while each of the robots were using an independent controller. When holding the objects, the robots coordinate their motion through direct sensing and communication. They can march in a tightly controlled formation while navigating autonomously. The experimental system containing three robots demonstrated the transportation of flexible boards and large boxes.

When a solid object is held by multiple robots, it also becomes a communication media where the force asserted by one robot can be estimated by other robots. [68] and [140] proposed a decentralized object transportation system featuring leader-follower coordination. Only the leader knew the high level information such as goal/path. The leader could either be a human being or a robot. The followers were homogeneous robots that were holding the object. Through estimating the motion of the leader by measuring force/moment from the object, a follower was able to assert both pulling and pushing forces on the object along the same direction as the leader's direction of manipulation.

Recently, aerial vehicles were designed to cooperatively lift and transport an object. [37] developed a cooperative aerial towing system using three quad-rotor aerial vehicles.

Pushing Systems

Transport by pushing does not require the robots to have mechanisms like grippers. In principle, pushing can be achieved by simply making a robot collide with the

object. In a cooperative transport scenario, the object is potentially larger than the robots. Therefore, it is unlikely that a robot can achieve a force closure of the object [115]. In fact, when the object is much larger than the robots, the contact between a robot and the object is virtually a point on the object's surface. In such a case, the object will rotate if the position of the pushing robot is not ideal. Even if the robot is able to determine a good position to push the object (e.g. based on knowledge about its geometry and mass distribution), the overall motion is still "generally unpredictable due to unknown support friction force." [70]. Therefore, the controller for pushing is difficult compared to the simple physical mechanism it requires. Furthermore, in such a case, a pushing robot can hardly assert any lateral force (in contrast to the normal force) to the object, which means the DOF of the actuation is smaller than the DOF of the object's motion. When the object is being pushed by multiple robots, it's rotation speed and translation speed can be controlled by adjusting the moving speed of each of the pushing robots.

In [74], a physical system that uses two six-legged robots to push a large rectangular object was presented. The object was an elongated rectangular box. It was moveable by one robot but the failure rate in the experiment was high. After using two robots with that cooperated their action through wired communication, the success rate of the transport could be substantially improved.

In [142], a physical experiment of cooperative box pushing was presented as a case study to verify a muti-robot system which used a host system to dynamically design and distribute homogeneous behavior-based controllers to the robots.

Cooperative pushing can also be achieved using very simple robots and behaviors. Kube *et al.* accomplished a series of works that involved pushing an object using a number of simple robots and behaviors [63][65]. Inspired by the behavior model of individual ants, the robots simply map the perceptual cues obtained from a small number of sensors onto nine motion primitives. Due to the simplicity of the control method, the number of robots working simultaneously in the cooperative transport task was flexible and a physical system containing 3 to 6 robots was used in the experiment.

In [148], a group of behavior-based robots pushed a transparent box towards a lamp in the environment without explicit communication. Each of the robots could select a suitable behavior set depending on the situation judged from its own sensors. For transporting a box, the behavior sets were divided according to

the number of robots nearby. In each set, the actions were a few discrete motion primitives selected according to the perception of the box, the light and nearby robots. The robots could also handle the situation where the light from the lamp was occluded by other robots.

In pushing-based cooperative manipulation, it is very common that the object needs to be specific so that it will not block the perception of those robots that are pushing it. In most of the works referred to above, the object needs to be transparent or lower than some of the sensors on the robots.

An alternative decentralized approach is through role differentiation using explicit inter-robot communication. For example, in [43], a box pushing system that contains robots with different roles is presented. In the case of cooperative transport, the roles are "pusher" and "watcher". The watcher is in front of the object so it can observe the goal while the pushers are behind the object. The robots communicate through WiFi.

In [55], an underwater box-pushing system is presented with three robotic fish; two of them work as pushers while the other works as an observer. The three fish can share sensing information through explicit communication to work out the approximate pose of the box, the two pushers can push on appropriate positions without seeing the goal directly.

Caging Systems

Cooperative transport by caging means to organize a group of robots into a formation around the object in a way that the object is caged inside the formation [98][118]. As long as the formation of the robots is maintained while they are moving, the object will follow the group of robots. Therefore, a caging systems do not need to deal with the complex dynamics of the object being pushed.

Caging an object using multiple robots is a problem similar to the problem of grasping with a robot hand — each of the robots is essentially a finger. Sudsang's research group presented some caging algorithm based on the grasping theories used in a robotic hand [118][119][96]. There many challenges to be overcome to implement caging based solutions in decentralized multi-robot systems. One reason is that much information is required to plan the caging cooperation. It is not necessarily possible that the robots are able to collect this information using

their own sensors.

In [113], a potential-field based controller was designed to transport an object by surrounding it.

Wang and Kumar proposed a caging algorithm based on object closure [141]. Rather than 'grasping' the object with the robots, a solution is employed that does not require the object to be in steady contact with all of the robots. As long as the object's origin point resides in an area called "closure configuration space", the object cannot escape the formation formed by the robots. The closure configuration space is an area bounded by the robots. To estimate it, each of the robots need to know the shape of the object and the position of the other robots. A series of papers had been published to exploit different aspects of the algorithm [137] [94] [138] [139]. In [95], a physical system based on this algorithm has been implemented using three robots. The controllers based on this algorithm were considered decentralized because all information was collected using the robot's own sensor. However, the information needed by this algorithm is still relatively large.

Fink *et al.* presented a physical caging system that copes with a variable number of robots [36]. In their system, the caging behavior contained three phases: approach, surround and transport. Within these three phases, surround is the phase where the robots form and stabilize a dynamic formation which orbits the object while avoiding collisions with other robots. Once the formation is stabilized, it will begin to shrink until some of the robots collide with the object. Then, the robots move toward the target location following a global navigation function or track a reference trajectory derived from the object's reference trajectory. In this behavior, the object was only pushed by one or two robots at a given time. This imposes a limit on the object's weight that can be handled by the system.

Swarm Robotic Systems

Since pushing is a manipulation that can be achieved by simple robots, it is suited for swarm robotic systems. However, one common point of pushing/caging based systems (including all of the systems referred to before except for [63] and [36]) is that the number of robots may not be large (typically not more than five). One important factor that limits the number of robots is the use of explicit communi-

cation to achieve highly cohesive behavior.

There are not many works that have studied how a relatively large group of robots can be used in a cooperative transport task when the controller only requires local information.

In [103], a physical system that includes up to 100 kilobots was used to study a decentralized strategy for collective transport. The strategy was evaluated in situations where the robots resided within the object being transported.

In [9], a large swarm of kilobots was controlled using a global input signal issued by a human operator to transport objects towards a goal.

2.4 Pattern Formation

2.4.1 In Nature

The spatial segregation of a group of relatively small entities often results in pattern formation. Pattern formations are widely existing in nature. Some of them are formed passively under external global affections from nature (e.g. sand dunes, Giants causeway). The kind of pattern formation considered here is self-organized pattern formation, which emerged from the active interaction of the cells/particles.

Granular Convection

The granular convection phenomenon is the spatial segregation of a massive number of particles of different sizes that occurs after shaking. The phenomenon is sometimes referred as the Brazil nut effect [101]. In [8], the Brazil nut effect is explained as follows:

> "During the periods when shaking loosens the packing, individual small particles can move into voids beneath large particles and so prevent them from returning to their previous positions. It is far less probable that several small particles will move together so as to create a void that can be occupied by a single large particle. The net effect is that the smaller particles occupy the lower positions during the active part of the shaking process and then become trapped there when the grains fix into a new arrangement."

[59] studied the particle segregation in 3-D. Their results show that the degree of segregation is "sensitive to both the size ratio and the number ratio for the case of a binary mixture of spheres."

Reaction-Diffusion Systems

A type of sophisticated pattern formation in nature can be found in the tissues of animals. The initial homogeneity of the animal embryo breaks down while developing. A well known theoretical model covering this phenomenon is the Turing's model [130]. His model suggested that such loss of homogeneity may be caused by a system of chemical substances reacting together and diffusing through the tissue. The difference in the propagation characteristics of different chemical substances across the tissue can introduce a pattern in the distribution of the chemical substances and thereby affects the biology of the cells. This model is also called the reaction-diffusion model.

The review paper in [61] summarized this theory and the reaction-diffusion model was promoted as follows:

> "The Turing, or reaction-diffusion (RD), model is one of the best-known theoretical models used to explain self-regulated pattern formation in the developing animal embryo. Although its real-world relevance was long debated, a number of compelling examples have gradually alleviated much of the skepticism surrounding the model. The RD model can generate a wide variety of spatial patterns, and mathematical studies have revealed the kinds of interactions required for each, giving this model the potential for application as an experimental working hypothesis in a wide variety of morphological phenomena." [61]

Despite that the model alone may have limitations in the reliability of the pattern formed [7], it can be used to describe to many self-organized patterns in biological systems. An example are skin patterns of animals, such as the black-white lateral pattern on zebra or the black-yellow pattern of cheetah.

Meinhardt *et al.* has published a number of works on theories of such biological pattern formations from a different perspective [77][60][75]. Despite the cells only reacting to the level of the chemical around themselves, they develop into a body with an apparent global polarity. One of the factors that caused such polarity is

the global gradient emerged from the long range nonlinear interactions of at least two chemicals and on their diffusion in the cells. Some of these patterns can even regenerate after being damaged. For example, the body of a hydra can recover the missing end when it is cut into two [44][76].

2.4.2 In Robotic Systems

A number of studies have looked at spatial segregation of a group of robots in computer simulation. For example, Şahin *et al.* [29] implemented a control law based on a probabilistic framework. Kumar *et al.* [66] implemented a control law based on artificial potential functions. In these studies, segregation is the result of "individual choices that discriminate" [106].

Segregation phenomena observed in ant colonies [40] have inspired the implementation of control laws for robots that organize two distinct groups of items into center-periphery patterns [147][78] (see also [1]).

Inspired from natural physical laws, [114] introduced an algorithm that uses artificial forces between the robots to organize the global formation of a swarm into a well arranged square or hexagon. The robots use only local information, which are the distances to the nearby robots and a binary state of a nearby robot. The paper also claimed that introducing noises in the system can improve the formation.

Ngouabeu *et al.* [87] observed segregation phenomena in a system of vibrating and non-vibrating mechatronic modules that float on the surface of water.

In [124], a control algorithm that can transform a circular formation into polygon formations or divide the swarm into three groups was proposed. To divide the swarm, some of the robots in the circular formation leave the formation while the rest of the robots keep executing the same algorithm that produces the circular formation. In [57], a robot can know its order in a periodic sequence in the circular formation through executing the Turing morphogenetic function. Thus, the robot can determine the required relative position to the adjacent robots to form a global polygon formation.

In [149], a distributed controller that can stabilize a naturally unstable formation brought by the asymmetrical attractions in a group of robots were studied.

In [122] proposed an object transportation method inspired by the Brazil nut

effect. In a space that contains a massive number of small robots and relatively large objects, the robots emulate the motion of the particles in the Brazil nut effect. Based on the effect, the objects will be moved along the direction opposite to the direction of global attraction applied on the robots. In the homogeneous version of this method, the global attraction was achieved through repulsion of the robots away from the destination. In the heterogeneous version of this method, two kinds of motion are propagated through the swarm. Both of the versions were verified in simulation experiments.

Other studies have looked at spatial segregation in the context of macroscopic self-assembly [49]. Bowden *et al.* [18] observed center-periphery structures when millimeter scale objects of two different heights interacted with each other by lateral capillary forces. In [16], an approach for self-reconfiguration of modular robots to create "emergent structures" with the desired functionality was achieved.

Chapter 3

Technological Platform

This thesis investigates how swarm robotic systems accomplish several tasks using both computer simulation and real environments. Many of the technological developments underpinning the implementation of these systems are similar across the tasks. For example, the computer simulations use the same simulation framework. Moreover, the physical robots use the same sensory framework, which has been further developed into a library published under the MIT license on GitHub [24].

This chapter presents the technological platforms used in this thesis.

3.1 Robot Platform

The robot used for physical implementation is the e-puck (see Fig. 3.1(a)), which is an off-the-shelf robot designed for education and research purposes[1]. It was developed by a collaboration among several laboratories in EPFL [82].

The e-puck is 7.0 cm in diameter, 5.5 cm high, and weighs 150 g. It is a differential-wheeled robot driven by two step motors. Its maximum linear and angular speed are 12.8 cm/s and 220 deg/s respectively.

The main microcontroller of the e-puck is dsPIC30F6014A. It has 8KB RAM, 144KB Flash ROM and sixteen 12-bit ADCs. The microcontroller is running at 58.9 Mhz, and has a processing capability of 14.7456 MIPS.

The e-puck has eight infrared proximity sensors distributed around its body. The proximity sensors are 3.1 cm above the ground. The robot also has a direc-

[1]The official homepage of the e-puck is `http://www.e-puck.org/`.

(a) (b)

Figure 3.1: The e-puck miniature mobile robot. (a) An unmodified e-puck. (b) An actual e-puck as used in this study. It is fitted with a black shell and a green top marker.

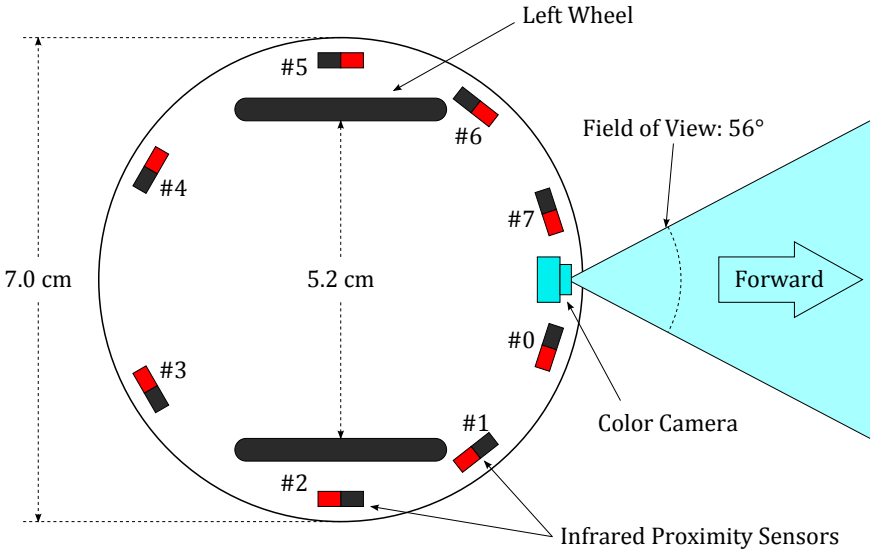


Figure 3.2: Top-view schematic of an e-puck, indicating the locations of its wheels, camera and proximity sensors. A detailed description of the hardware of e-puck can be found in [82].

tional color camera in the front of its hull, pointing towards the front. The camera is 2.8 cm above the ground. On top of the e-puck, there is an infrared receiver. It can decode the modulated infrared signal from a TV remote. Around its "ring", there are eight red LEDs that can be controlled separately.

Fig. 3.2 shows a schematic of the e-puck from above, including the locations of the sensors used in this thesis.

3.1.1 Physical Modifications

The e-puck design has some shortcomings that may affect its application in a swarm robotic system:

1. The color of an e-puck is mainly gray. Thus, in an experimental scenario with a light-colored background, it is difficult for the e-puck's onboard camera to identify another e-puck effectively.

2. There are some red decals around the e-puck's body. Red is frequently used as the color of important clues in visual perception. For example, a red cylindrical object is used as the goal in the cooperative transport study presented here, while the red LEDs on the e-pucks are used for signaling in the fully decentralized segregation study. The red decals on the e-puck may cause misperception of such visual clues.

3. The infrared proximity sensors on an e-puck can hardly detect another e-puck. Due to its mechanical design and building material, the e-puck does not have an ideal infrared reflection property. In most situations, the ability of one e-puck to detect other e-pucks greatly depends on their relative positions.

4. When viewing a group of e-pucks inside a arena from above, the top of an e-puck can easily induce specular reflection of the main light source in the environment. This can cause problems in robot tracking algorithms, even during offline processing.

To deal with problems (1), (2) and (3), each of the e-pucks was fitted with a black shell made of paper. By doing so, in the camera image from the e-puck, other e-pucks are clearly distinguishable against the light-colored environment.

Table 3.1: Working Configuration of The Camera on e-puck

Resolution	320×240 (Interlaced)
Color Format	RGB565
Exposure Time	Fixed at 1.0×
Gain	Fixed at $1.0, 1.0, 1.0$
White Balance	Fixed at $1.125, 1.0, 1.4375$
Resulting Frame Rate	≈ 18.4 fps

The black shell does not reflect infrared light at all. So the controller for the robots can then be designed assuming the robots are not infrared reflectors. The detection of nearby e-pucks was realized by changing the working mode of the proximity sensors, which will be introduced in detail in Section 3.1.3.

To deal with (4), top markers made of card paper were placed over the e-pucks. In the segregation study, the robots in different groups have tops with different colors so the spatial segregation can be visualized.

Fig. 3.1(b) shows a photograph of one of the actual e-pucks used in this research. The printable files of these materials can be found in [24].

3.1.2 Camera and Image Processing

The e-puck has a camera[2] that is directly wired to its main micro-controller. The camera is configured through the I2C bus while the pixel data are transferred byte-by-byte through 8 digital pins. The configuration used in this research is given in Table 3.1.

The camera is configured to send a QVGA interlaced video stream. In this mode, each of the incoming frame has a resolution of 320×120. The 120 rows in the odd frames are all odd rows while the 120 rows in the even frames are all even rows. Therefore, two of such frames can be composed into one full image with a resolution of 320×240.

The micro-controller on the e-puck is not capable of processing the 320×120 frames in real time due to limited memory and clock rate. Therefore, the frames are sub-sampled to 40×15 pixels and each of them is considered as an individual

[2]The camera model of the e-pucks used in this research is PO6030K.

Figure 3.3: What an e-puck "sees" when it is facing another e-puck nearby.

frame. As a result, the 40×15 pixels in the image are all refreshed when each of the frames in the odd-even frame pairs is captured. This means that the refresh rate of the camera images is the same as the refresh rate of the video stream provided by the camera, which is ≈ 18.4 fps. The sub-sampling is implemented by taking one pixel every 8 pixels from the original frames. Therefore, the field-of-view is not affected significantly and there are no filtering algorithms applied.

Fig. 3.3 shows an image transmitted back from an e-puck when it is facing another e-puck nearby.

In this thesis, the on-board camera on the e-puck was used to perceive the existence and/or the relative angle of any entity within a range that is relatively long (e.g. 1.5 m) compared to the size of an e-puck. To extract these information, each of the captured images is processed to provide four scalar values: (i) the number of pixels that are considered red and blue, denoted by c_r and c_b respectively, and (ii) the horizontal distribution biases of the red and the blue pixels, denoted by e_r and e_b respectively. The algorithm to extracts these scalar values from image is detailed in the following.

Let R_{xy}, G_{xy} and B_{xy} denote the red, green, blue components of pixel (x, y) respectively.

$F_{red}(x, y)$ determines whether the color of pixel (x, y) is considered as red:

$$F_{red}(x, y) = \begin{cases} 1 & \text{if} \quad R_{xy} > 0.4 \quad \text{and} \quad \frac{R_{xy}}{G_{xy}} > 1.4 \quad \text{and} \quad \frac{R_{xy}}{B_{xy}} > 1.3 \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

$F_{blue}(x, y)$ determines whether the color of pixel (x, y) is considered as blue:

$$F_{blue}(x, y) = \begin{cases} 1 & \text{if} \quad B_{xy} > 0.4 \quad \text{and} \quad B_{xy} - R_{xy} > 0.1 \quad \text{and} \quad B_{xy} - G_{xy} > 0.1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

The constants in the equations above were manually tuned to work well in the environment used for experiments in this thesis.

The rest of the steps are described using the process on blue pixels as a instance. These steps are similar for the process on red pixels.

The number of blue pixels, c_b, is given by summing $F_{blue}(x, y)$ over x and y:

$$c_b = \sum_x \sum_y F_{blue}(x, y), \tag{3.3}$$

The horizontal bias of the blue pixels, e_b, is horizontal center of "mass" of the image when the "mass" of the blue pixels and the non-blue pixels are considered 1.0 and 0.0 respectively. Let w be the width of the image. If $c_b > 0$, e_b is calculated as

$$e_b = \left(2 \times \frac{\sum_x \sum_y x F_{blue}(x, y)}{w c_b}\right) - 1. \tag{3.4}$$

Otherwise, e_b is set to 0.

In (3.4), e_b is clamped to $[-1, 1]$ where negative and positive values correspond to a left and a right bias, respectively.

The image processing of the camera is implemented using a dual buffer swapping mechanism, in which the image acquisition and the image processing are concurrently working on two different buffers. In this mechanism, as long as the image processing does not take more time than the image acquisition, the results of the image processing algorithm can be updated as frequent as the image acquisition FPS ($\approx$ 20 times per second in this case).

3.1.3 Proximity Sensors

The eight infrared proximity sensors on the e-puck are able to provide approximate distance to objects with a range around 8.0 cm. Each of these sensors is composed of one infrared emitter and one infrared transistor. The eight infrared emitters are connected to four digital pins of the microcontroller, so they can be switch on and off in groups of two. The eight infrared transistors are connected to eight ADC pins independently.

In order to satisfy the sensing requirements of the experiments in this research, different working modes for these components had been developed. Through changing the working scheme of the components of the proximity sensors, three working modes had been developed:

- "EMITTER OFF" mode (same as the e-puck standard library)
- "EMITTER ON" mode
- "EMITTER NOISE" mode

The first two modes will be described in the following sections. The third working mode is not used in any of the systematic experiments (but only a demonstration) discussed in this thesis, so it will only be introduced briefly.

"EMITTER OFF" mode

This working mode follows the conventional working scheme of infrared proximity sensors[3]. In this mode, the infrared emitters will be turned on only when the sensors are going to be read. If the infrared light from the infrared emitter is reflected by an object, the infrared transistor along side that emitter will detect an increase in infrared intensity. This working mode is widely used in the detection of the presence of an object and the approximate distance to that object.

Proximity sensors working in the conventional mode cannot detect objects if their surface does not reflect infrared light effectively. As explained in Section 3.1.1, the e-puck is not a good reflector for infrared light. To overcome this problem, another working mode was implemented.

"EMITTER ON" mode

In this working mode, the e-pucks keep their infrared emitters turned on. As a result, the proximity to another e-puck can be determined through the ambient infrared intensity. The detailed implementation is as follow.

The eight infrared emitters are turned on when the robot is initialized. Then, a sampling procedure will be executed once every 50 ms. In the beginning of a sampling procedure, the eight infrared transistors are sampled immediately. Because the emitters on the robot are kept turned on beforehand, these samples are the sum of the ambient and the reflected infrared intensities. The eight values can be represented as a vector:

$$\mathbf{M} = \begin{bmatrix} m_0 & m_1 & \dots & m_7 \end{bmatrix}^{\mathbf{T}}. \tag{3.5}$$

[3]This mode is also the one used by the standard e-puck library.

The subscripts of each element in this matrix corresponds to the id of the proximity sensor, which can be seen in Fig.3.1(b).

Then, the emitters are turned off. After a 300 μs delay, the transistors are sampled again. These results represent only the ambient intensities:

$$\mathbf{A} = \begin{bmatrix} a_0 & a_1 & \dots & a_7 \end{bmatrix}^{\mathbf{T}}. \tag{3.6}$$

The emitters are then turned on again, in preparation for the next sampling procedure.

The reflected intensities are calculated as the difference between the combined and the ambient intensities, i.e.: $\mathbf{R} = \mathbf{M} - \mathbf{A}$, where

$$\mathbf{R} = \begin{bmatrix} r_0 & r_1 & \dots & r_7 \end{bmatrix}^{\mathbf{T}}. \tag{3.7}$$

The whole procedure of the sampling steps is given in Algorithm 1.

Algorithm 1 "EMITTER ON" mode sampling procedure.

```
Initialization (runs once at the start):
    Turn on the 8 IR emitters;

Sampling procedure (runs every 20 ms):
    M = Sample the 8 IR transistors;
    Turn off the 8 IR emitters;
    Wait for 300 us;
    A = Sample the 8 IR transistors;
    R = M − A;
    Turn on the 8 IR emitters.
```

Most of the experiments in this thesis used a large number of e-pucks (e.g. at least 20). The start signal of an experimental trial was broadcast from a TV remote. Thus, the e-pucks should receive this signal simultaneously. Once the start signal is received, the e-puck acquires a random seed. This random seed is based on the proximity sensor readings and the standby duration from booting up, therefore, it is likely to be unique for each of the e-pucks. This random seed is used to keep the sampling routine of proximity sensors across the robots asynchronous. Combined with the fact that the OFF period of the infrared emitters is negligible

(300 μs every 20 ms), each of the robots is effectively a constant source of infrared radiation. For this reason, a robot is able to distinguish between other robots and passive items (e.g. objects or walls). A positive ambient intensity corresponds to the presence of nearby robots, while a positive reflected intensity corresponds to the presence of a nearby passive item. The proximate distance to a robot or a passive object can be estimated from these intensities. The range of the intensities is scaled to $[0, 1]$. The higher the intensity, the closer is the source.

After extensive development, it is possible to utilize the proximity sensors to achieve explicit communication [125][103][99]. This is however not considered here due to the scope of the research.

The "EMITTER ON" mode can not be applied in an environment with strong ambient infrared light, for example, in a room illuminated by day light or incandescent lamps. In such an environment, the ambient samples ($\mathbf{A}$) cannot be used to determine the proximity to other robots.

"EMITTER NOISE" mode

The "EMITTER NOISE" mode was implemented in this research for a demonstration of the cooperative transport in an environment lit mainly by day light. In this mode, the infrared emitters are constantly emitting noise in the form of a series of spikes with random intervals between them. By ignoring the constant level of infrared light, a robot can determine the existence and approximate distance to any nearby robots through the amplitude of the spikes in the level of infrared light perceived.

This mode has been used in a demonstration of the cooperative transport system in a domestic environment, which will be introduced in Section 4.9.

Note that the infrared receiver (the one used to receive a global start signal from a TV remote) can be severely interfered by the proximity sensors working under this mode, which may make it difficult to control the start of an experimental trial. Therefore, this working mode has not been applied in any experiments for scientific purpose.

3.2 Simulation Framework

The experiments in this thesis use physical e-puck robots as the implementation platform. However, to prepare a large number of physical robots is very time-costly. For example, the microcontrollers on the e-pucks can only be programmed one by one manually through a tedious Bluetooth pairing procedure. This was especially problematic when the development was in its early stage because modifications need to be applied frequently. Furthermore, the systematic behaviors of a swarm robotic system are often stochastic. It is often necessary to do repeated trials to achieve the verification and analysis of the system. Therefore, a simulation framework that is fast and simple enough for swarm robotics simulation is important.

There are several robotic simulators available as freeware or commercial software. Webots[4] is one of the most widely used commercial simulator in robotics. The robots in Webots have a very detailed physical and graphical model. This means the accuracy of the simulation could be high when there are a small number of robots. However, its high computation cost makes Webots not suitable for simulating a scenario that may involve dozens of robots. Stage[5] is a multiple-robot simulator that can be used to simulate a scene with hundreds of robots in real time [131]. However, its collision detection is based on discretized space and is not accurate enough for interactions like object pushing. Thus, a simulation framework that is optimized for our research was developed.

The simulation framework is based on the Enki simulator[6]. Enki is a fast 2-D simulator that can handle 2-D rigid body physics and sensor interactions. It is used by Webots in 2-D mode. Moreover, it has a pre-built model for the e-puck. However, Enki's physics simulation was found to be not accurate enough in some scenarios, for example, when multiple robots are pushing a large object cooperatively. Our solution is to override the rigid body physics using the Bullet Physics library[7]. The simulation framework is also able to render a visualization of the simulation in real time. Fig. 3.4 shows a screenshot of a simulation built upon the simulation framework. The rest of this section will introduce the simulation

[4] `http://www.cyberbotics.com`
[5] `http://playerstage.sf.net`
[6] `http://home.gna.org/enki`
[7] `http://bulletphysics.org`

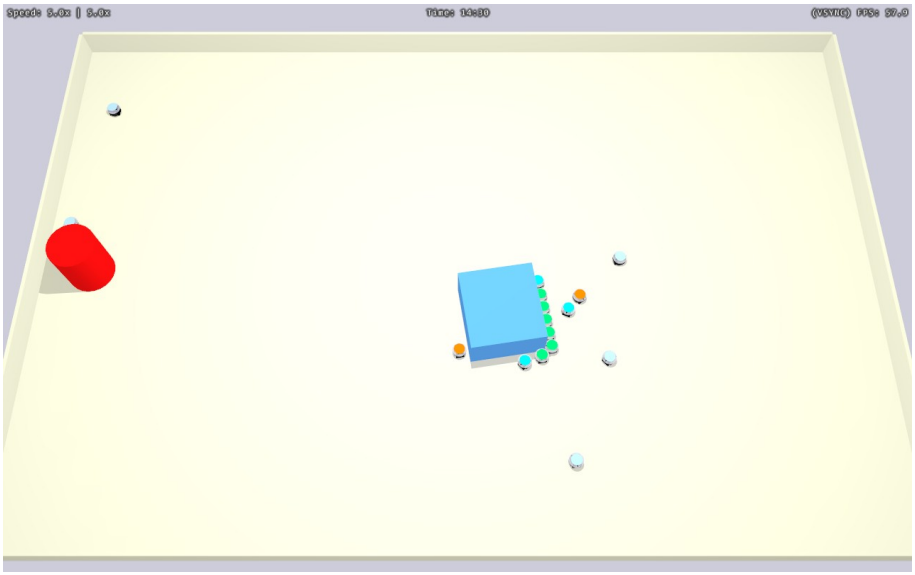


Figure 3.4: A screenshot of the simulator when the robots are running the cooperative transport controller studied in Chapter 4.

framework.

3.2.1 Incorporating Bullet Physics Engine

In the simulation framework, the physics simulation is processed by Bullet Physics. To simulate the e-puck robot using Bullet Physics, a 3-D physics model was built according to the e-puck's mechanical design. This model can be found in Appendix A.1.

The two stepper motors on the e-puck are simulated as two DC motors. In Bullet Physics, this can be achieved by making the hinge constraint representing the wheel shaft a DC motor. One can also specify the desired rotation speed of such a DC motor directly. Therefore, the wheel speeds in the simulation framework can be set in a very similar way to those on the real e-pucks.

The proximity sensors and camera of the e-puck are still simulated by Enki. In Enki, the simulation of the proximity sensors is accurate enough for this research. However, the image provided by the camera is simulated as a one dimensional image. This image is equivalent to the center row of the camera image on the real e-puck. In other words, the camera images are 40×1 pixels in the simulation while it is 40×15 on the real e-pucks.

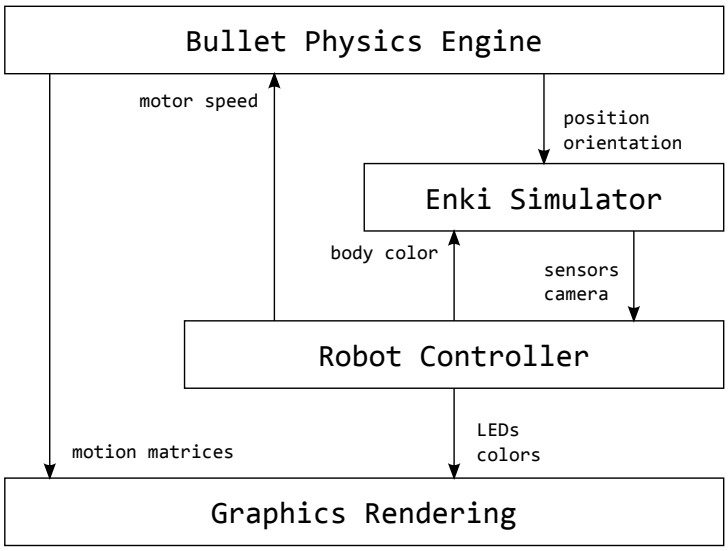


Figure 3.5: This diagram illustrates the role of each component in the simulation framework by showing the data provided and taken by the components.

Simulating sensor interaction generally requires knowledge of the positions of the robots. In the original Enki simulator, the positions of the robots are computed using Enki's built-in 2-D physics engine. In the simulation framework here, the position of a robot is obtained from its physics model in Bullet Physics and then passed to its physics model in Enki. Because Bullet Physics is a 3-D physics engine, it provides positions as 3-D vectors. Only the x term and y terms are passed to the Enki simulator. This may not cause a significant difference in the simulation of a planar environment when the object can only rotate along the z-axis.

Fig. 3.5 illustrates the relation between the different components within this simulation framework.

3.2.2 Expanding Proximity Sensors Simulation

The proximity sensors on the e-puck are simulated by Enki. However, Enki is only capable of calculating the output of the proximity sensors assuming that they are working in the conventional mode (the "EMITTER OFF" mode). As introduced in previous sections, the "EMITTER ON" mode is important for the swarm robotic system studied here. Therefore, the simulation of the "EMITTER ON" mode of the proximity sensors were developed.

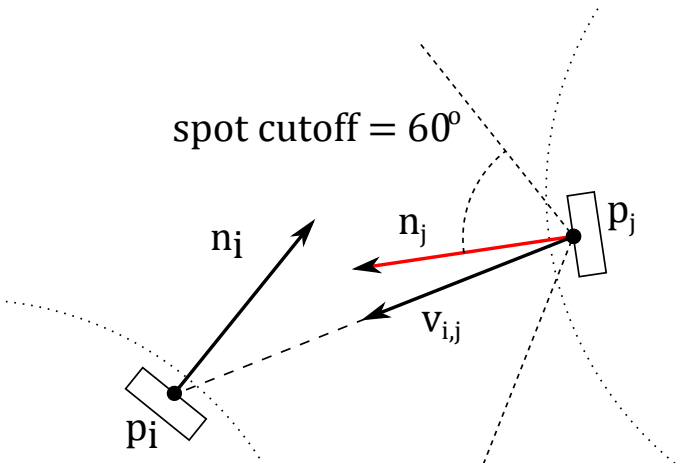


Figure 3.6: This diagram shows the vectors that are involved in the calculation of the infrared lighting from the infrared emitter in sensor j to the infrared transistor in sensor i.

In the "EMITTER ON" mode, the proximity sensors on the e-puck are always on when it is not sampling the ambient infrared intensity. As a result, the infrared light emitted by any e-puck can be easily detected by its nearby e-pucks. To simulate this interaction, the infrared light emitted by e-pucks' infrared emitters needs to be calculated for each of the proximity sensors in the scenario.

Because the properties of infrared light are almost identical to visible light, the mathematical modeling of this interaction is based on a lighting model used in computer graphics. The particular model used is derived from the Phong reflection model. In this case, the output of a proximity sensor is based on diffusive intensity at the position of the sensor. The detailed mathematical description of this calculation is presented in the following.

A typical infrared proximity sensor consists of one infrared emitter and one infrared transistor. Inside such a sensor, the positions and normals (direction vector) of the two components are approximately identical. For sensor i, its position and normal vector are denoted $\mathbf{p}_i$ and $\mathbf{n}_i$ respectively.

In "EMITTER ON" mode, the ambient intensity[8] output of a sensor is contributed to by the emitters of other sensors nearby. Therefore, the ambient inten-

[8]Note that: this ambient intensity is not the ambient intensity in the Phong reflection model. For a proximity sensor, the ambient infrared intensity is the intensity of infrared light that is not sourced from the sensor's own infrared emitter (via reflection).

sity output of sensor i can be formulated as follows:

$$q_i = \sum_j I_{i,j}. \tag{3.8}$$

In the above equation, $I_{i,j}$ is the contribution of infrared light intensity of sensor j to sensor i. Fig. 3.6 illustrates the above definitions.

The contribution of one sensor to another sensor (function $I(\cdot)$) is calculated through several steps. The first step is to compute the distance and the normalized displacement vector between the two sensors. They are denoted as $d_{i,j}$ and $\mathbf{v}_{i,j}$ respectively and defined as follows:

$$d_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\|. \tag{3.9}$$

$$\mathbf{v}_{i,j} = \frac{\mathbf{p}_i - \mathbf{p}_j}{d_{i,j}}. \tag{3.10}$$

The infrared emitter inside sensor j is modeled as a spot light with a 120 deg coverage angle. The lighting effect of a spot light is related to the dot product of its normal and the displacement vector. This dot product is often called the "spot dot". A spot dot is larger when the point to be lit is nearer to the normal axis of the spot light. In the case here, the spot dot is calculated as:

$$w_{i,j} = \mathbf{n}_j \cdot \mathbf{v}_{i,j}. \tag{3.11}$$

The spot dot is also used to judge whether the position is within the effective cone (defined by coverage angle) of the spot light. In the case here, when the angle between $\mathbf{v}_{i,j}$ and $\mathbf{n}_j$ is larger than 60 deg ($w_{i,j} < \cos \pi/3$), sensor i is considered to be outside the effective cone of sensor j.

If sensor i is not in the effective cone of sensor j, the rest of the steps will be skipped while the result ($I_{i,j}$) is constantly zero.

A spot light is a type of directional light. Therefore, its intensity on any point on a surface is proportional to the dot product of the displacement vector and the normal vector of the surface on that point. This dot product is

$$b_{i,j} = -\mathbf{n}_i \cdot \mathbf{v}_{i,j}. \tag{3.12}$$

Distance attenuation is also considered, which means that the farther sensor i is from sensor j, the weaker is the intensity. It is calculated as:

$$a_{i,j} = \|d_{i,j}\|^{-1.733}. \tag{3.13}$$

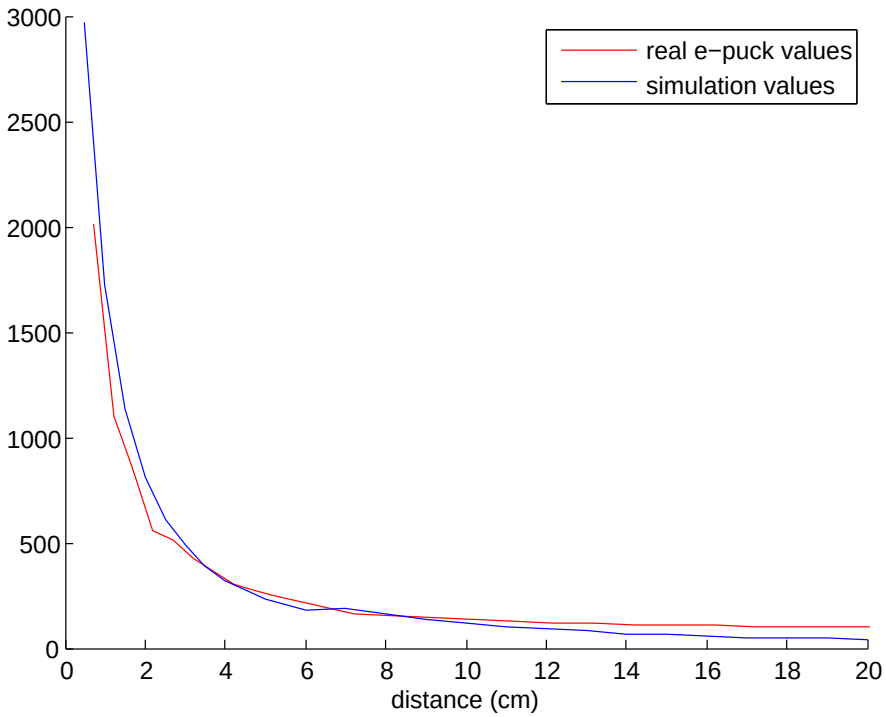


Figure 3.7: Infrared ambient intensity of proximity sensor 2 on an e-puck in simulation and a real e-puck.

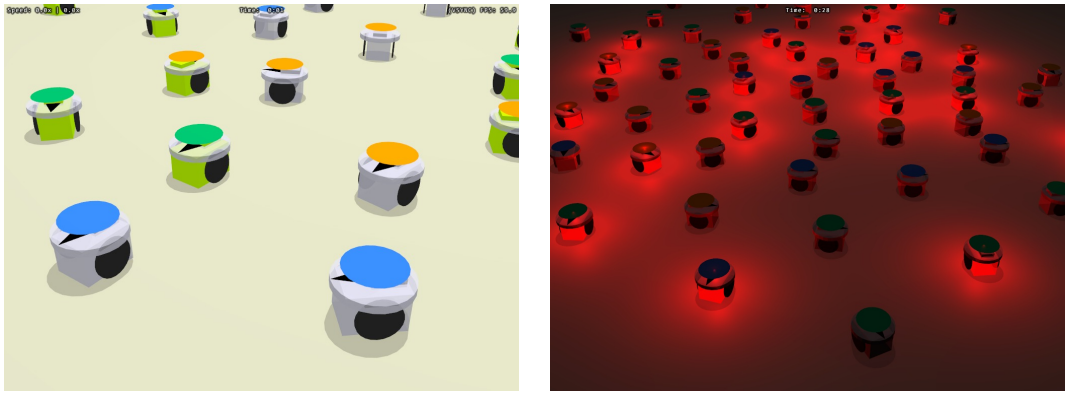
The power index of -1.733 is estimated by fitting parameters of the sensor output curve of the real e-pucks.

Finally, the contribution of sensor j to sensor j is

$$I_{i,j} = kw_{i,j}b_{i,j}a_{i,j}. \tag{3.14}$$

In (3.14), k is a constant gain of 5000 that scales the output value to an approximate range of $(0, 4000)$, which corresponds to the range used on real e-pucks.

Fig. 3.7 compares the reading from a real e-puck against the reading from an e-puck in simulation. In both testing environments, an e-puck in "EMITTER ON" mode was placed side-by-side by a distance to the measuring e-puck. Then, the ambient sample of the proximity sensor 2 (on the right hand) was recorded. The distance between the two e-pucks was measured between the two rings on the two e-pucks.



(a) (b)

Figure 3.8: Screenshots of the simulation framework, showing the graphic model of the e-puck. The geometry primitives used to construct e-puck's graphic model approximate the e-puck's physics model (see Appendix A.1), which means the rectangular box and the 'disk', as well as the two wheels, have physical embodiments. (a) A swarm of e-pucks with caps of different colors (b) Visualization of the ring LEDs on the e-pucks.

3.2.3 Graphic Rendering

The simulation framework is also capable of rendering the simulation scene in real time on screen. The graphics system is developed based on OpenGL. In Fig. 3.8(a), the graphics model of the e-puck can be seen.

To enhance the visibility of the moving direction of an e-puck, an black arrow marker and a small rectangular box (without physical body representation) are drawn above the ring of the e-puck. An overhead maker can be placed on top of an e-puck when necessary. In simulation, the color of this marker can be changed dynamically for debug purposes, for example, showing the internal state of an e-puck.

Because the studies in this thesis do not require the 8 ring LEDs to be turned on or off individually, these LEDs are rendered as one LED. To increase the visibility of the LEDs, they are rendered as light sources in the scene. A customized GLSL rendering pipeline was written to handle upto 64 light sources simultaneously. The lighting effect of these LEDs can be clearly seen when the main light source in the scene is disabled (see Fig. 3.8(b)). This visual enhancement played an important role in the simulation study of the decentralized segregation (see Section 6.4.1).

Chapter 4

Occlusion-Based Cooperative Transport

This chapter introduces a strategy for transporting a large object to a goal using a large number of mobile robots that are significantly smaller than the object. The robots only push the object at positions where the direct line of sight to the goal is occluded by the object. This strategy is fully decentralized and requires neither explicit communication nor specific manipulation mechanisms. This strategy has been implemented on the e-puck robotic platform in both computer simulations and a real environment. A computer simulation in a 3-D environment has also been presented using a conceptual robot. The simplicity of this strategy makes it particularly well-suited for micro-scale robotic systems.

4.1 Problem Formulation

The task that we consider is as follows. A bounded environment contains a convex-shaped object, a goal, and a number of robots. The environment is otherwise free of obstacles. The aim is that the robots, which are initially placed in arbitrary locations, transport the object to the goal. Note that the goal specified in the problem may not be the final destination of the transportation. In a broader scenario, the goal could be moving, or it could be one of a series of way points [89]. This scenario will be introduced later.

We make the following assumptions. The dimension of the object is large

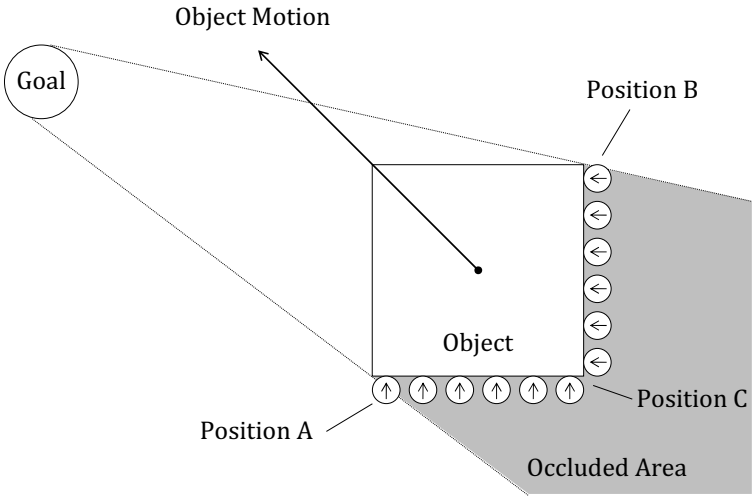


Figure 4.1: Illustration of how a swarm of robots can push a large object in a 2-D planar environment. The robots keep pushing only along the section of the object's perimeter that occludes their views of the goal. As a consequence, the motion of the object will be approximately towards the goal.

enough to occlude the robots' perception of the goal when they are behind it (see Fig. 4.1). The robots can perceive the goal from any point within the environment, unless it is occluded by the object. The robots can only move the object by pushing it. The robots are able to distinguish between the object, each other and the boundaries of the environment.

4.2 Strategy Description

Consider a number of robots that can distribute themselves uniformly around the section of the object's surface that occludes their view of the goal (the "back side" of the object), as shown in Fig. 4.1. Then, if all the robots push the object by moving in a direction perpendicular to the object's surface at their points of contact, the overall motion of the object will be approximately towards the goal. As the object moves, its occluded surface changes over time, thus changing the direction of motion. If the robots keep pushing *only* against the occluded surface, the object will eventually reach the goal. Note, the strategy could in principle be also used for transporting objects that are not tall enough to occlude the robots'

view of the goal. If a robot reached the object but the goal was visible 'behind' it, the robot would then still push.

The occlusion-based transport strategy can be realized using a fully decentralized controller and without explicit communication among the robots. In Fig. 4.2, the behavior of the individual robots is given in form of a state machine. A robot first searches the object using an algorithm that is suitable for the environment ('*Search Object*'). For bounded environments, as considered in this study, the robot performs a random walk. More sophisticated search algorithms could help the strategy to also cope with unbounded environments. Once the object is seen, the robot moves towards it while avoiding other robots ('*Approach Object*'). When the robot has reached the object, it enters state '*Check for Goal*' to work out whether the goal can be seen from its position. If the goal can not be seen, the robot will push the object simply by moving against it ('*Push Object*'). If the goal can be seen, the robot will attempt to find another position around the object ('*Move Around Object*'), for example, executing a left-hand-following behavior.

When executing the state machine, the robots eventually end up at different positions along the occluded section of the object due to the stochastic nature of the system [107]. However, the more robots that are used, the more likely it is that they approximate a uniform distribution (as shown in Fig. 4.1).

4.3 Mathematical Analysis

In this section, we analyze the occlusion-based cooperative transport strategy for the case of arbitrary convex objects in planar environments. We prove that, under some idealized assumptions, the strategy always succeeds in moving the object to the goal. Note that the transport strategy is not suited for objects of *arbitrary* concave shapes (for a counter example, see Section 4.3.4).

4.3.1 Modeling of the Occlusion Problem

We assume that each of the goals and robots are points (without embodiment). Let $\mathbf{c} \in \mathbb{R}^2$ be the center of mass of a rigid convex object with respect to a coordinate frame in which $\mathbf{g} = [0, 0]^T$ is the goal point. Let the perimeter of the object be

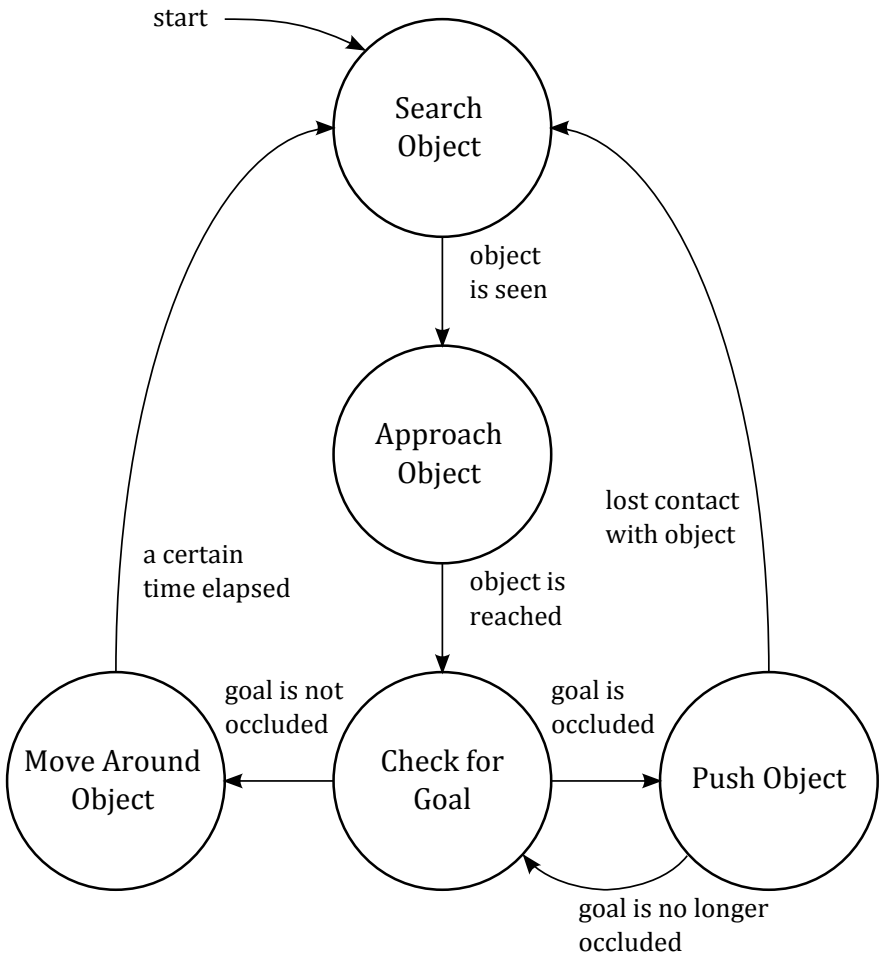


Figure 4.2: A state machine representation of the controller summarizing the core behaviors of a robot in the strategy. The start state is 'Search Object'. If the object is lost during any state, the robot will restart from 'Search Object'. The controller is fully decentralized and does not require any explicit inter-robot communication.

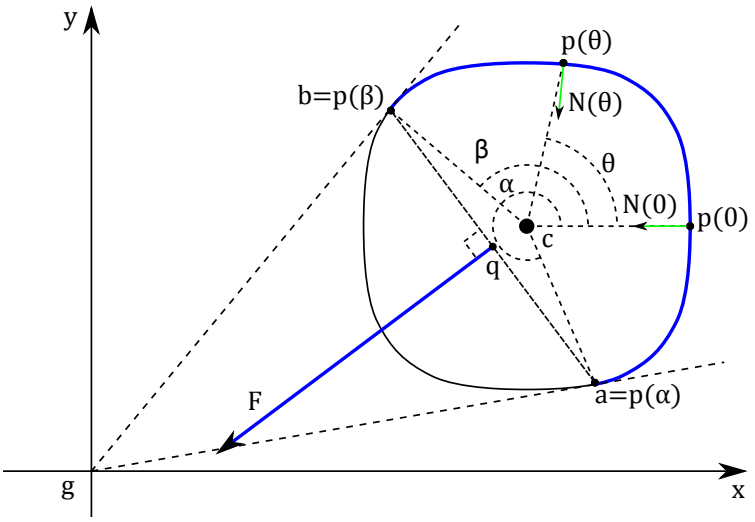


Figure 4.3: If normal forces are uniformly applied on the blue section of the convex-shaped object's perimeter (major arc **ab** in this diagram), the combined force vector, **F**, is the vector $(\mathbf{b}-\mathbf{a})$ rotated by $+\frac{\pi}{2}$ and its magnitude is proportional to the length $\mathbf{b}-\mathbf{a}$ (chord **ab** in this diagram). Point **q** is an affecting point of **F**.

described by a closed, convex and differentiable curve given by:

$$\mathbf{p}(\theta) = \begin{bmatrix} r(\theta)\cos\theta \\ r(\theta)\sin\theta \end{bmatrix} + \mathbf{c}, \tag{4.1}$$

with $\theta \in [0, 2\pi]$ and $r : [0, 2\pi] \to \mathbb{R}$ differentiable and satisfying $r(2\pi) = r(0)$. By specifing $r(\theta)$, any convex shape can be approximated by $\mathbf{p}$. Initially, $\mathbf{g}$ is outside $\mathbf{p}$.

The inwards pointing normal vector on $\mathbf{p}(\theta)$, named $\mathbf{N}(\theta)$, is the derivative of $\mathbf{p}(\theta)$ rotated by $\frac{\pi}{2}$:

$$\mathbf{N}(\theta) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{p}'(\theta). \tag{4.2}$$

Points along $\mathbf{p}$ where the direct line of sight to $\mathbf{g}$ is occluded are between the two tangent points of $\mathbf{p}$ from point $\mathbf{g}$. We write the two tangent points as $\mathbf{p}(\alpha)$ and $\mathbf{p}(\beta)$, $\alpha, \beta \in [0, 2\pi]$. As tangent points, they satisfy:

$$\begin{aligned} &\mathbf{p}(\alpha) \cdot \mathbf{N}(\alpha) = \mathbf{p}(\beta) \cdot \mathbf{N}(\beta) = 0, \\ &\mathbf{p}(\theta) \cdot \mathbf{N}(\theta) > 0, \forall\theta \in (\alpha, \beta). \end{aligned} \tag{4.3}$$

Since $\mathbf{p}$ is convex and $\mathbf{g}$ is outside $\mathbf{p}$, α and β are well defined. For convenience, write $\mathbf{a} = \mathbf{p}(\alpha)$ and $\mathbf{b} = \mathbf{p}(\beta)$. Additionally, they are named so $\mathbf{a}$ is the tangent point on the right side of vector $(\mathbf{c} - \mathbf{g})$ while $\mathbf{b}$ is the one on the left side. Strictly speaking, $\mathbf{a}$ and $\mathbf{b}$ satisfy

$$\begin{aligned} a_x c_y - a_y c_x > 0, \\ b_x c_y - b_y c_x < 0, \end{aligned} \tag{4.4}$$

with x and y subscripts denoting the x and y coordinates. These properties of $\mathbf{a}$ and $\mathbf{b}$ will play an important role in the proof of the transport strategy later.

Fig. 4.3 illustrates the above definitions. In colloquial terms, all points $\mathbf{p}(\theta)$ with $\theta \in (\alpha, \beta)$ are on the occluded perimeter of the object while all other points on p are visible from $\mathbf{g}$.

The Resultant Force Applied on The Object

Lemma 1. *Assume that $n \to \infty$ robots are uniformly distributed along the occluded perimeter of the object and they are the only robots asserting a force on the object. The direction of the resultant force asserted on the object by the robots is equal to the direction ofthe vector $(\mathbf{b} - \mathbf{a})$ rotated by $+\frac{\pi}{2}$ and its magnitude is proportional to $\|\mathbf{b} - \mathbf{a}\|$.*

Proof. According to the strategy, all robots along the occluded perimeter assert normal forces on $\mathbf{p}$. Without loss of generality let the magnitude of the force be one unit force per unit length. The combined force is the definite integral given by

$$\mathbf{F} = \int_{\alpha}^{\beta} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{p}'(\theta)\, \mathrm{d}\theta. \tag{4.5}$$

The solution of the definite integral in (4.5) is:

$$\mathbf{F} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \Big(\mathbf{p}(\beta) - \mathbf{p}(\alpha) \Big), \tag{4.6}$$

which is:

$$\mathbf{F} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (\mathbf{b} - \mathbf{a}). \tag{4.7}$$

□

We can also derive the torque around the z-axis caused by the robots. For this, with slight abuse of notation, we interpret all previous points as embedded in the x, y plane in $\mathbb{R}^3$. Again, we assume that the magnitude of the force is one unit force per unit length. Then the magnitude of the torque around z-axis contributed by all robots with respect to point $\mathbf{c}$ is

$$Q = \int_{\alpha}^{\beta} [(\mathbf{p}(\theta) - \mathbf{c}) \times \mathbf{N}(\theta)] \cdot \hat{\mathbf{z}} \, \mathrm{d}\theta, \tag{4.8}$$

where $\hat{\mathbf{z}}$ represents a unit vector pointing along the z-axis. The part within the integral is equal to

$$\begin{bmatrix} r(\theta)\cos(\theta) \\ r(\theta)\sin(\theta) \\ 0 \end{bmatrix} \times \begin{bmatrix} -r'(\theta)\sin(\theta) - r(\theta)\cos(\theta) \\ r'(\theta)\cos(\theta) - r(\theta)\sin(\theta) \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{4.9}$$

which can be simplified to $r'(\theta)r(\theta)$. Then (4.8) can be written as:

$$Q = \int_{\alpha}^{\beta} r'(\theta)r(\theta)\mathrm{d}\theta. \tag{4.10}$$

Its solution is:

$$Q = \frac{r^2(\beta) - r^2(\alpha)}{2}. \tag{4.11}$$

Lemma 2. *If the combined force contributed by the robots,* $\mathbf{F}$*, is considered as a single force while* Q *is the torque induced by* $\mathbf{F}$*, the mid point of segment* $\mathbf{ab}$ *is an affecting point of* $\mathbf{F}$*.*

Proof. Naming the affecting point of $\mathbf{F}$ as $\mathbf{q}$, $\mathbf{F}$, $\mathbf{q}$ and Q must satisfy

$$Q = [(\mathbf{q} - \mathbf{c}) \times \mathbf{F}] \cdot \hat{\mathbf{z}}. \tag{4.12}$$

The above equation can be transformed into

$$\mathbf{q} \cdot (\mathbf{b} - \mathbf{a}) = \frac{r^2(\beta) - r^2(\alpha)}{2} + \mathbf{c} \cdot (\mathbf{b} - \mathbf{a}), \tag{4.13}$$

which can be viewed as the vector equation of a line.

While $\mathbf{q}$ can be any point on (4.13), we make $\mathbf{q}$ a convenient point on (4.13), which is

$$\mathbf{q} = \frac{\mathbf{a} + \mathbf{b}}{2}. \tag{4.14}$$

□

4.3.2 Motion Dynamics of the Object

As the object is moved, $\mathbf{a}$ and $\mathbf{b}$ can change over time. We assume that the robots react instantly to such changes so that the occluded perimeter is always uniformly filled up with pushing robots. Thus, (4.7) is valid at any point in time as long as $\mathbf{g}$ is outside $\mathbf{p}$. In other words:

$$\mathbf{F}(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \Big(\mathbf{b}(t) - \mathbf{a}(t)\Big). \tag{4.15}$$

From (4.15), it follows that the rotation of the object does not affect the relationship between $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{F}$. According to Newton's laws, the translation dynamics of the center of mass of the object are

$$\mathbf{v} = \dot{\mathbf{c}},\ \dot{\mathbf{v}} = \frac{\mathbf{F}}{M}, \tag{4.16}$$

where $\dot{\mathbf{v}}$ (respectively $\dot{\mathbf{c}}$) is the derivative of $\mathbf{v}$ (respectively $\mathbf{c}$) with respect to time t.

We can apply a quasi-static analysis to the case here in which some robots are pushing a rigid object slowly [69]. Then the translation dynamics of the object is

$$\dot{\mathbf{c}} = k\mathbf{F}, \tag{4.17}$$

where $k \in \mathbb{R}^+$ is a positive constant that transfers $\mathbf{F}$ proportionally to the velocity of the object.

4.3.3 Convergence of the Object's Distance to the Goal

Theorem 1. *The distance between* $\mathbf{c}$ *and* $\mathbf{g}$ *is strictly decreasing over time if the velocity of the object is governed by* (4.17). *As* $t \to \infty$, $\mathbf{g}$ *will be on the object perimeter* $\mathbf{p}$.

Proof. Let $l(t) = \mathbf{c}(t) \cdot \mathbf{c}(t)$ be the squared distance of the center of mass $\mathbf{c}$ to goal $\mathbf{g}$, then its derivative with regard to time is

$$\dot{l} = 2k\mathbf{c} \cdot \mathbf{F}. \tag{4.18}$$

Substituting $\mathbf{F}$ with (4.7), we get

$$\mathbf{c} \cdot \mathbf{F} = (b_x c_y - b_y c_x) - (a_x c_y - a_y c_x). \tag{4.19}$$

According to (4.4), $\mathbf{c} \cdot \mathbf{F} < 0$. Hence, $l(t)$ is strictly decreasing. Since $l(t) \geq 0$ for all $t > 0$ (as long as $\mathbf{g}$ is outside $\mathbf{p}$), we get $\lim_{t\to\infty} l(t) = L \in \mathbb{R}$. Therefore,

$$\lim_{t\to\infty} \mathbf{c} \cdot \mathbf{F} = \lim_{t\to\infty} b_x c_y - b_y c_x + a_y c_x - a_x c_y = 0, \tag{4.20}$$

which together with (4.4) implies that:

$$\begin{aligned} \lim_{t\to\infty} b_x c_y - b_y c_x = 0, \\ \lim_{t\to\infty} a_y c_x - a_x c_y = 0. \end{aligned} \tag{4.21}$$

In other words, the areas of the triangles **gca** and **gcb** approach zero as $t \to \infty$. Since **c** is always inside **p** the triangles **gca** and **gcb** can never have 0 area unless $\mathbf{a} = \mathbf{g}$ and $\mathbf{b} = \mathbf{g}$ (see Fig. 4.3). Hence as $t \to \infty$, **g** will be on **p**. In other words, the object will ultimately coincide with the goal and stop moving.

□

4.3.4 A Counter Example for Concave Objects

The mathematical analysis has proven that the combined force introduced by the transport strategy can always reduce the distance between a convex object and the goal. Such result is not true for some extreme concave objects depending on the relative distance and orientation of the object to the goal. For instance, Fig. 4.4 shows a counter example with $\mathbf{c} \cdot \mathbf{F} > 0$. In other words, the resultant force asserted by all robots will move the object away from the goal.

4.4 Implementation for e-puck

Section 4.2 described the individual behavior required to achieve this transport strategy in a platform-independent form. Based on the frameworks introduced in Chapter 3, this individual behavior is implemented as a fully decentralized controller for the e-puck robotic platform.

4.4.1 State-machine Implementation

In Section 4.2, the core behaviors of this transport strategy are given in form of a state-machine. For the e-puck robots, the strategy is also implemented as a state-machine.

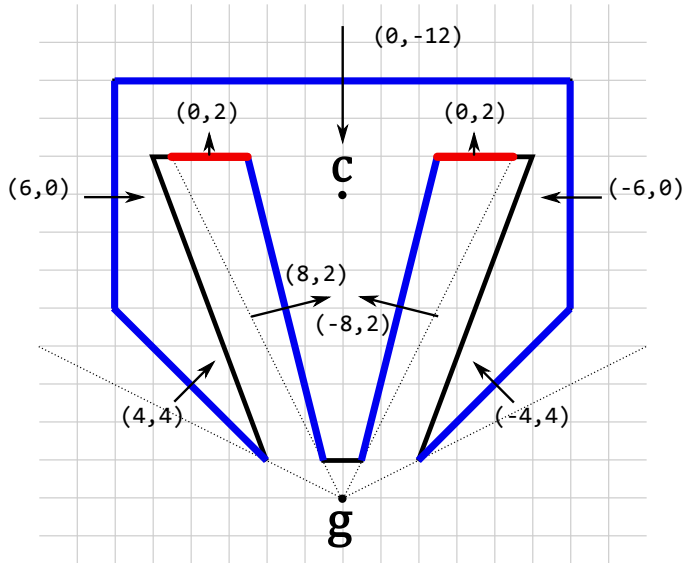


Figure 4.4: On the perimeter of this concave object, both red and blue segments are occluded from the goal. The coordinates in the drawing are the forces brought by each of the segments measured in grid units. The combined force brought by the robots at the blue segments is zero, whereas the combined force brought by the robots at the two red segments pushes the object away from the goal.

The robot performs a random walk and approaches any blue object seen in its camera. If the robot loses sight of the object, it resumes the search. When it reaches the object, it does a full rotation to look for the red goal. If the goal is not seen, the robot starts pushing the object. If the goal is seen, the robot executes a left-hand-wall-following behavior, which relocates the robot to a position where the goal may be occluded by the object.

When in the pushing formation, a robot's perception of the goal may not only be occluded by the object but also by its neighboring robots (or potentially the environment boundaries). However, the robots at the two ends of the formation (i.e. at Positions A and B in Fig. 4.1) can effectively monitor the visibility of the goal. For the e-pucks with the sensing mechanism introduced in Section 3.1.3, such a position could be judged from the inputs of the IR sensors on the left-hand and right-hand side: if both give high values of ambient intensity, then the robot has two neighbor robots. These robots can be considered as observers. When an observer perceives the goal, it leaves the formation. As a consequence, its neighbor becomes an observer. Thus, those pushing robots that are no longer in

the occluded perimeter happen to leave in a recursive manner. For e-pucks, this behavior is utilized so that only observers are required to scan the environment for the goal while the other pushing robots can be devoted exclusively to pushing the object.

In order to make the strategy work in a real environment, basic behaviors like collision avoidance and error handling are added. The detailed state-machine implemented on the e-puck is given in Fig. 4.5. The states are:

- **S1: Search Object.** The robot moves randomly while avoiding collisions with other robots, the goal, and the boundaries of the environment. Its linear speed and angular speed are randomized every 5 seconds.

- **S2: Move to Object.** The robot moves towards the object using the camera output e_b while avoiding other robots.

- **S3: Close In on Object.** The robot closes the gap between itself and the object. In this state, the robot is 'attracted' to the object and 'repelled' from other robots, such that it eventually comes into contact with the object's perimeter.

- **S4: Scan and Align.** The robot does a full rotation on the spot to scan for the goal using its directional camera. While rotating, the robot also adjusts the distance to its neighbors to achieve a relatively uniform arrangement.

- **S5: Push Object.** The robot pushes the object by moving perpendicularly to the object's surface in front of it. To increase the efficiency, only those robots that have less than two neighbor robots are required to scan for the goal after pushing for 3 seconds.

- **S6: Move Around Object.** The robot performs a wall-following behavior to move around the object and other robots that are near the object (e.g. pushing it). By doing so, the robot searches a point around the object's perimeter that is valid for pushing.

- **S7: Evade.** The robot moves away from anything nearby, avoiding collisions.

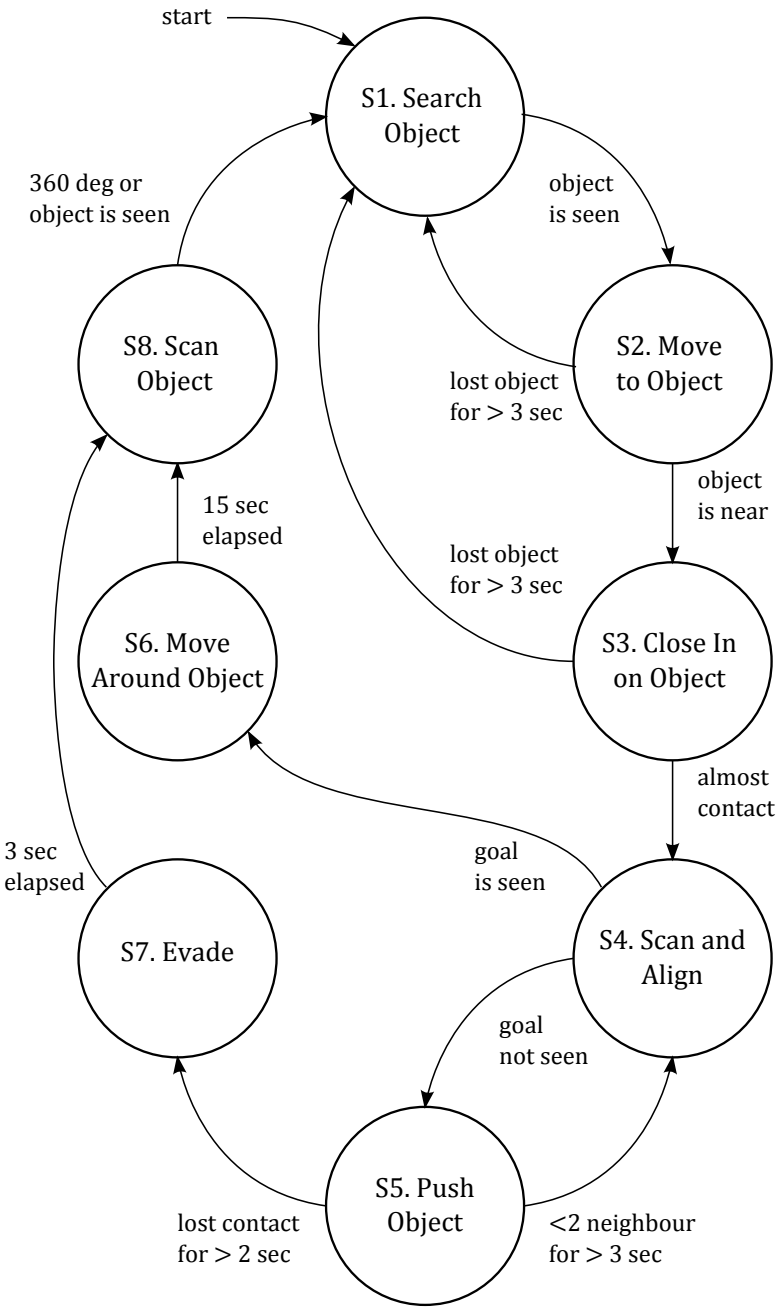


Figure 4.5: The full state machine implementation of the strategy on the e-puck robots, including some behaviors to increase the robustness and handle errors. The initial state is S1: Search Object.

- **S8: Scan Object.** The robot rotates to scan for the object using its directional camera.

The state transition conditions are shown in Fig. 4.5. For most of the conditions, certain sensory inputs are compared against a preset threshold. In each of the states, specific low-level motion controllers are activated to achieve the required motion. The motion controllers for these behaviors are described in the next section.

4.4.2 Low-level Motion Control

The e-puck's motion is controlled by setting the rotation speeds of its left and right wheels. These speeds are denoted by ω_l and ω_r, and are normalized within the interval $[-1, 1]$, where -1 and 1 correspond to the maximum backward and forward rotation speeds of the wheels, respectively. They are calculated according to:

$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1 & -s \\ 1 & s \end{bmatrix} \begin{bmatrix} S \\ D \end{bmatrix}, \tag{4.22}$$

$$s = \begin{cases} 1 & \text{if } S \geq 0, \\ -1 & \text{if } S < 0. \end{cases} \tag{4.23}$$

In (4.22), S represents the linear speed of the robot, whereas D represents its angular speed and both are within the interval $[-1, 1]$. For S, -1 and 1 represent the maximum backward speed and maximum forward speed of the robot. For D, -1 and 1 represents the maximum clockwise angular speed and maximum anticlockwise angular speed of the robot. The values of ω_l and ω_r given by (4.22) are saturated to $[-1, 1]$ if they lie outside this range.

The values of D and S are calculated according to:

$$D = \mathbf{D_A A} + \mathbf{D_R R} + n_D + k_e e_b, \tag{4.24}$$

$$S = \mathbf{S_A A} + \mathbf{S_R R} + n_S, \tag{4.25}$$

where $\mathbf{A}$, $\mathbf{R}$ and e_b are the variables from the sensors (as described in Section 3.1.3), and the other values are the state-dependent parameters:

- k_e is a scalar that weights e_b. It is used to generate an angular speed bias to turn the robot towards the object in relevant states.

Table 4.1: The parameters configuration of the motion controllers.

	n_S	n_D	k_e	$\mathbf{D_A}$		$\mathbf{D_R}$		$\mathbf{S_A}$		$\mathbf{S_R}$	
				R.	L.	R.	L.	F.	B.	F.	B.
S1. Search Object	0.85	δ	0	U	U	U+	U+	N	0	N	0
S2. Move to Object	0.8	0	-0.4	U+	U+	U	U	N	0	N	0
S3. Close In on Object	0.6	0	-0.4	U+	U+	T	T	N	0	N+	0
*S5. Push Object **	0.55	0	0	U	U	T	T	N	0	N	0
S6. Move Around Object	0.5	0.5	0	0	U	0	U+	P	0	P	0
S7. Evade	0	0	0	U	U	U	U	N+	N+	N+	N+
S8. Scan Object	0	0.5	-0.3	0	0	0	0	N	N	N	N

Weight Property: T: Attractive, U: Repulsive, P: Positive, N: Negative, 0: Null ("+" sign means relatively strong)

Sensor Number: R: #0~#3, L: #4~#7, F: #0, 1, 2, 5, 6, 7, B: #3, 4

*Note: ***S5. Push Object*** does not follow the generic rule strictly

- $\mathbf{S_A}$ and $\mathbf{S_R}$ are 1×8 matrices that contain the weighting factor of $\mathbf{A}$ on S and $\mathbf{R}$ on S respectively. Depending on the signs and magnitudes of these weights, they can accelerate or decelerate the robot relative to a nearby perceived source. We call such an effect the *property* of a weighting matrix. The properties of $\mathbf{S_A}$ and $\mathbf{S_R}$ can be either 'Positive' (accelerate) or 'Negative' (decelerate).

- $\mathbf{D_A}$ and $\mathbf{D_R}$ are 1×8 matrices that contain the weighting factor of $\mathbf{A}$ on D and $\mathbf{R}$ on D respectively. They determine how the nearby robots and passive reflectors will affect the angular speed of the robot. Their properties can be either 'Attractive' or 'Repulsive', which turns the moving direction of the robot towards or away from the perceived source respectively.

- n_D and n_S are scalars that represent the neutral values on D and S respectively. These two neutral values give the ideal angular and linear speeds required to achieve the motion in the corresponding states.

The properties of the four state-dependent weight matrices were selected according to the desired motion in the corresponding state, so the coarse weights were determined. Then, the weights, as well as other constant parameters in the controller, were manually refined in pilot experiments using physical e-puck robots. Table 4.1 shows a summary (in terms of properties) of the state dependent

parameters in (4.24) and (4.25). The entire set of parameters can be found in [25].

For 6 out of 8 states in the state machine, the motion controller followed the rule described above. The two states that did not strictly follow the rule will be introduced next.

For S4: Scan and Align

In this state, D is fixed to 0.35, such that the robot always rotates with a constant angular speed in order to observe its surroundings. When the robot is rotating, it will control its distance to any robots in front of it by applying a linear speed S of magnitude 0.2 when the object is on either the left or the right side of itself. Otherwise, S is set to 0. The pseudo code of this program is given in Algorithm 2.

Algorithm 2 Speed Control for **S4: Scan and Align**

```
S = 0.0
IF a_0 > 0.04 AND a_7 > 0.04 THEN
    IF r_2 > 0.08 OR r_5 > 0.08 THEN
        IF a_0 < 0.15 AND a_7 < 0.15 THEN
            S = 0.2
        ELSEIF a_0 > 0.3 OR a_7 > 0.3 THEN
            S = −0.2
        ENDIF
    ENDIF
ENDIF
```

For S5: Push Object

In this transport strategy, the pushing robots move perpendicularly towards the object's surface in front of it. If the object has a curved perimeter (e.g. circle), this means that the distance between two pushing robots will become smaller when the object starts moving. Thus, collisions between the robots in the pushing formation will occur. This problem is magnified by the e-puck's design: two e-pucks will easily get stuck when they collide. Thus, the motion controller used the pushing state absolutely needs to deal with the collision avoidance problem.

From the physical experiments in the preliminary version of the controller published in [26], we found that when a robot is in contact with the object, the attraction of the object (the $\mathbf{D_R R}$ term in (4.24)) is very strong so that the repulsion of

Table 4.2: The weight parameters used by the motion controller in state "S5: Push object".

Sensor	#0	#1	#2	#3	#4	#5	#6	#7
$\mathbf{D_A}$	1.5	1.125	0.75	0	0	-0.75	-1.125	-1.5
$\mathbf{D_R}$	-1.5	-1.125	-0.375	0	0	0.375	1.125	1.5
$\mathbf{S_A}$	-0.25	-0.125	0.0	0	0	0.0	-0.125	-0.25
$\mathbf{S_R}$	0.0	-0.125	-0.25	0	0	-0.25	-0.125	0.0

a neighbor robot (the $\mathbf{D_A A}$ term in (4.24)) cannot effectively turn the robot away from the neighbor robot. Therefore, an extra rule is applied to the design above. When a nearby robot is too close (judging by the intensities in $\mathbf{A}$), the robot will only focus on avoiding that robot, which means the object attraction is nullified. Otherwise, the robot only focuses on moving perpendicular to the object, which means the repulsion from other robots is nullified. Algorithm 3 gives the details of the angular speed motion controller used in S5: Push Object. The speed (S) calculation remains as shown in (4.25).

Algorithm 3 Angular Speed Control for **S5: Push Object**

```
IF r_0 + r_7 > 0.6 THEN
    IF a_1 > 0.25 OR a_2 > 0.5
        OR a_5 > 0.5 OR a_6 > 0.25 THEN
        D = D_A A + n_D
    ELSE
        D = D_R R + n_D
    ENDIF
ELSE
    D = D_A A + D_R R + n_D
ENDIF
```

4.5 Experiments with Objects of Different Shapes

After analyzing the transport strategy mathematically, we obtained an indication of objects with not-unusual shapes that are nevertheless challenging for the strategy to handle. In this section, a set of experiments is introduced to evaluate the

strategy using objects of these shapes as well as compare the experiments against the prediction from the mathematical model.

4.5.1 Experimental Setup

Objects

Three objects of different shapes and sizes were used:

1. **A circular object** Theoretically, this is an ideal case as the resulting force points directly to the goal. However, in practice, the curved perimeter could make the robots more prone to collide with each other as the object is being moved. Such a situation may also occur for a rectangular object. For example, in Fig. 4.1, Position C is a point where all the pushing robots taxis towards when the object is moving. Therefore, it is essential that the collision avoidance mechanism in the pushing state is effective. As the pushing forces of e-pucks are rather limited, at least three robots are required to push this object.[1]

2. **A scalene triangular object** This is a simple example of a non-symmetric object. In this case, the ratio of the lengths of the triangle's sides is 3:4:5. According to Lemma 2, the robots cannot push this object along a straight line, because the resultant force vector will never pass through the object's centroid (i.e. the resultant torque can never be zero). As a result, depending on which side(s) the robots are pushing from, the object will rotate either clockwise or anticlockwise. Two robots pushing on the same side near the sharpest corner are enough to rotate this object. On the other hand, it takes at least four robots pushing on the same side in order to induce a translational motion.

3. **An elongated rectangular object** This shape is problematic for the occlusion-based transport strategy, because the resultant force can deviate by almost 90 degrees from the ideal direction of transport. The object easily rotates if the pushing formation is not uniform; in fact, one robot pushing at

[1]Depending on the floor condition and robot power, occasionally this object may also be pushed by just two robots.

Table 4.3: summary of the experimental setup and data

Object Shape		**Circular**	**Triangular**	**Rectangular**
Physical Property				
Size (cm)		diameter: 40.0	side: $45 - 60 - 75$	side: 58.5×13.5
Height (cm)		10	14	6.5
Mass (g)		222	432	160
Pushing Force (N)		0.75	1.5	0.5
Summary of Results				
Successful Trials		15 out of 15	14 out of 15	14 out of 15
Completion Time (sec)	mean	220.0	255.1	295.4
	σ	26.3	63.0	183.1
Path Efficiency	mean	0.914	0.793	0.766
	σ	0.029	0.099	0.192
AE (deg)	mean	26.7	90.1	204.6
	σ	16.8	36.2	79.2

one end is sufficient to induce a rotation. It takes at least two robots pushing on the same side in order to give a translational motion to this object.

The physical details of the three objects are given in Table 4.3. The mass of each of the objects was chosen so that it is theoretically possible for the e-pucks to push the object from all directions. The side of the objects are painted blue. Two orange markers of different size are attached on the top of each object, so that its position and orientation can be tracked in an off-line analysis.

Environment

The environment of the experiment is a rectangular arena of size 400×225 cm that is bounded by 50 cm-high walls. The floor of the arena has light gray color, and its walls are painted in white. The goal is a red cylinder of 25 cm diameter and 42 cm height, as shown in Fig. 4.7.

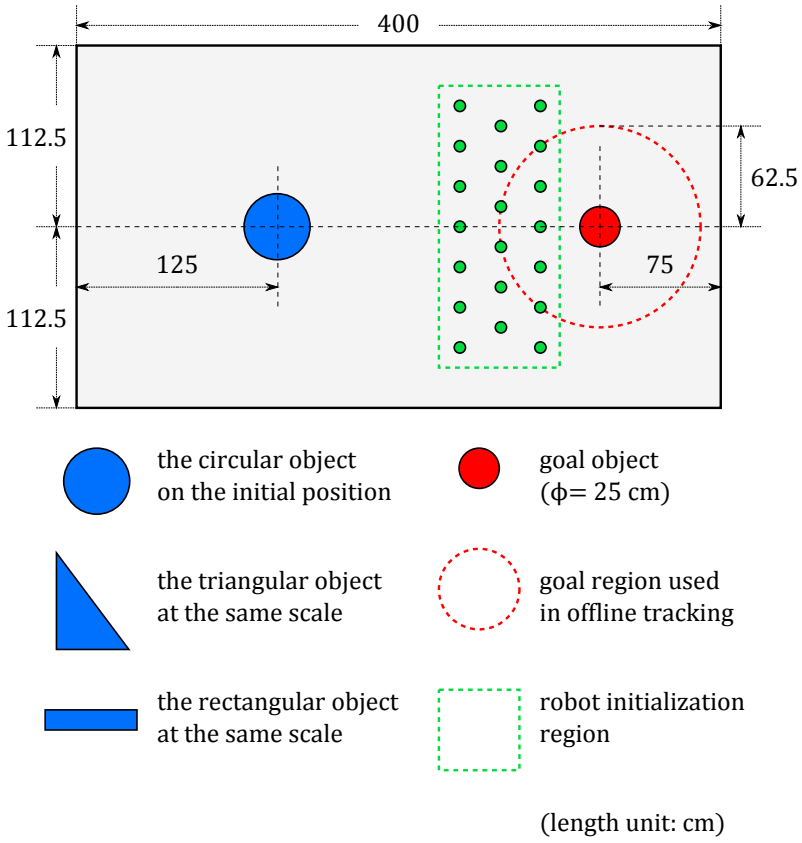


Figure 4.6: Initial positions configuration of the experiment. The robots were placed approximately in such a formation because the self-calibration of the proximity sensors on the e-puck requires a certain amount of space around the robot.

Trial Procedure

For each of the objects, we conducted 15 trials (45 trials in total). The number of robots used in each trial was 20. This was much larger than the least number of robots required for pushing the objects because the scalability of the strategy benefits from the use of more robots when dealing with objects of various sizes and shapes.

The initial configuration of a trial is illustrated in Fig. 4.6. The object's centroid was positioned as indicated. The orientation of the object was generated using a random number generator. The robots were placed in a zone between the object and the goal. The actual positions of the robots were loosely snapped to a grid to ensure a minimum gap between robots which is required by our self-calibration

routine for the e-puck. Before starting a trial, each robot rotated by a random proportion of a full rotation to obtain its initial orientation. The trials were started by issuing a signal via an infra-red remote control that is received by all of the robots simultaneously. The robots were programmed to stop automatically after 15 minutes.

A trial was stopped if either of the following situations happened:

1. The object collided with the goal object. The trial was then considered successful.

2. All of the robots stopped automatically due to the 15-minute time limit. This means the trial was unsuccessful.

3. The object was too close to the wall and thus cannot be transported via pushing any more. For example, either side of the triangular object fully touched the wall. This means the trial was unsuccessful.

The trials were recorded with an overhead camera. The videos were used in the off-line tracking of the object. The accompanying video shows three experimental trials, one for each type of object, respectively. Videos of all the 45 trials are available in [25].

4.5.2 Results and Discussions

Successful Trials

Overall, 43 out of the 45 trials were successful. The object reached the goal within 15 minutes. One trial with the triangular object failed. The other failed trial was with the rectangular object. In both cases, one side of the object became very close to the boundaries of the arena. This was due to the limited width of the arena and a relatively large error in the transport direction.

Completion Time

The completion time, T_k, is defined as the time elapsed from the start of a trial until the centroid of the object is less than 62.5 cm away from the center of the goal (i.e. when the centroid of the object is within the goal region in Fig. 4.6).

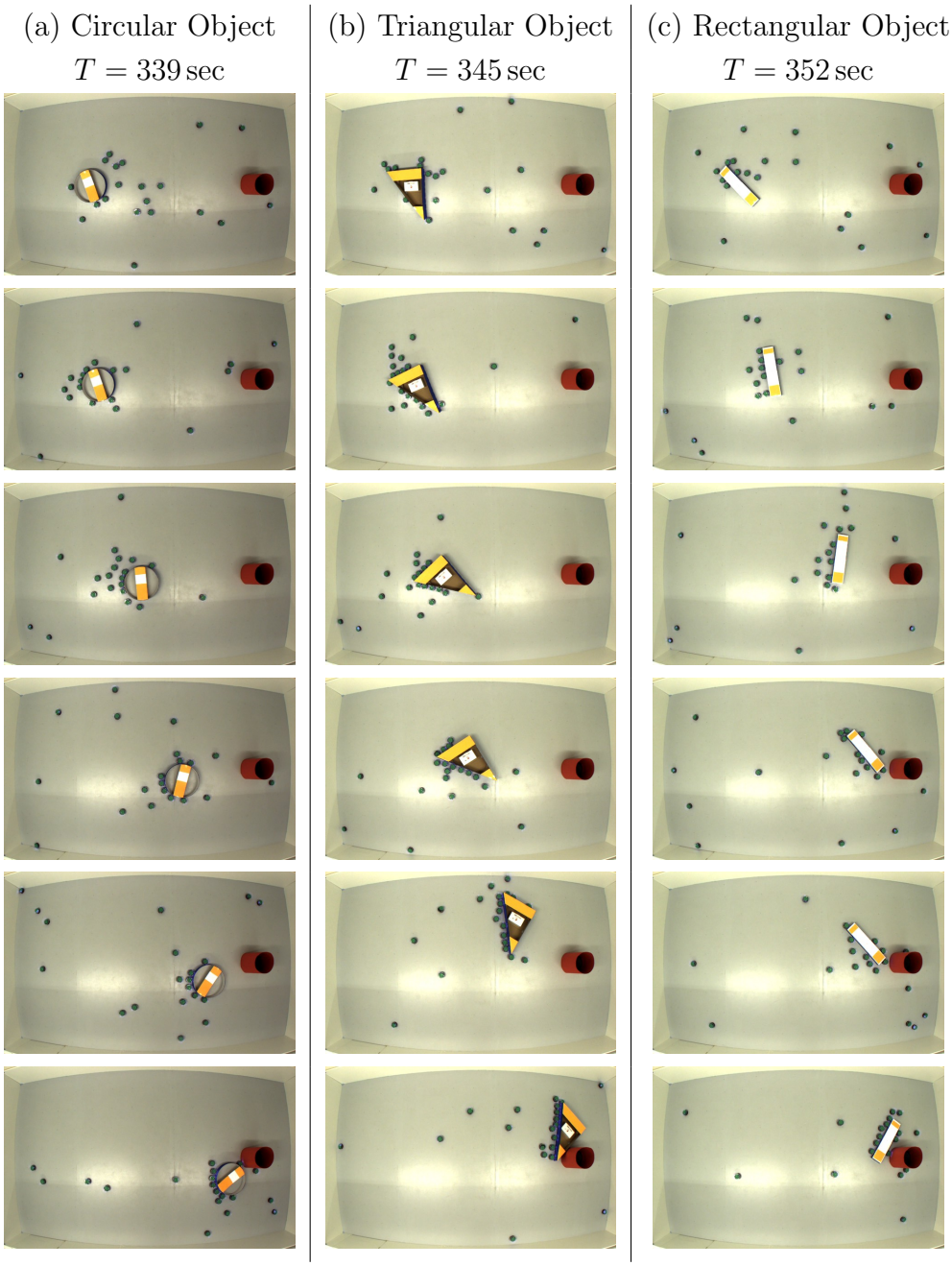


Figure 4.7: Snapshots showing three trials with similar durations in the systematic experiments with circular, triangular and rectangular object respectively. T is the total length of the videos (in seconds), which ends at the moment when the object collides with the goal. Videos of all the 45 trials are available in [25].

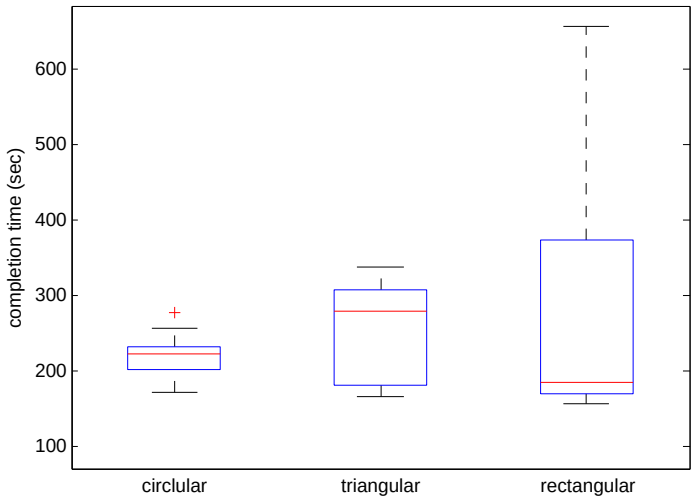
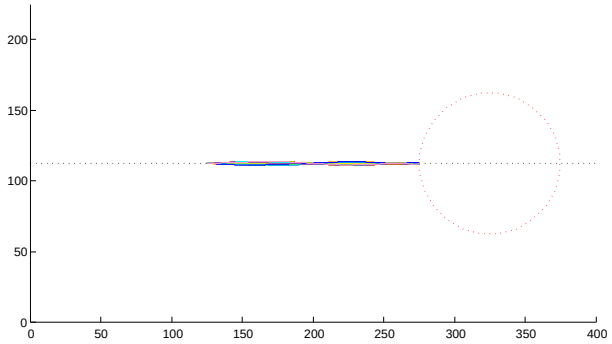


Figure 4.8: Completion time of the circular object, scalene triangular object, and elongated rectangular object.

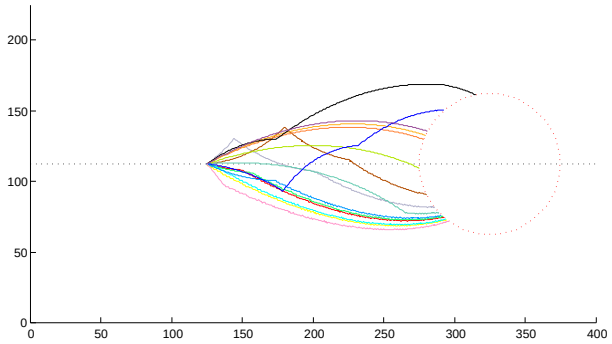
A box-and-whisker plot[2] of the completion time is given in Fig. 4.8. The deviation of the completion times of the triangular object and the rectangular object are larger than the circular object, which shows that the shape of the object will affect the transport. One case easy to be observed in the trials is that: if the elongated rectangular object reaches an orientation with either of its two small sides pointing towards the goal, it cannot be pushed effectively anymore. In Fig. 4.7(c), it can be observed from the last three snapshots that such a situation stalled the transport by more than 60 seconds. It depends on randomness when the robots manage to rotate the object out of such situation.

Object Paths According to Lemmas 1 and 2, the resultant force and torque applied on the object can be calculated given the initial position and orientation of the object and goal position (assuming an infinite number of point robots are equally dispersed around the occluded perimeter of the object). When the force and torque are directly transferred to the velocity and angular velocity of the

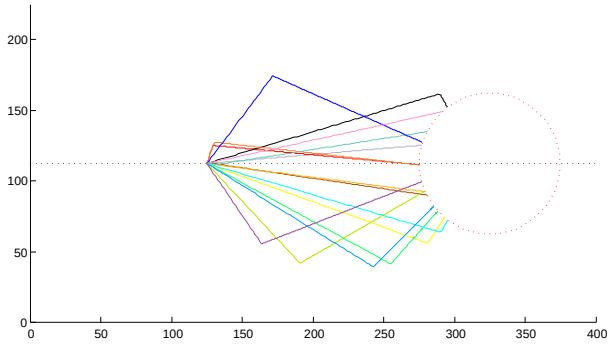
[2]The line inside the box represents the median of the data. The edges of the box represent the lower and the upper quartiles (25-th and 75-th percentiles) of the data, while the whiskers represent the lowest and the highest data points that are within 1.5 times the inter-quartile range from the lower and the upper quartiles, respectively. Crosses represent outliers.



(a) Circular Object

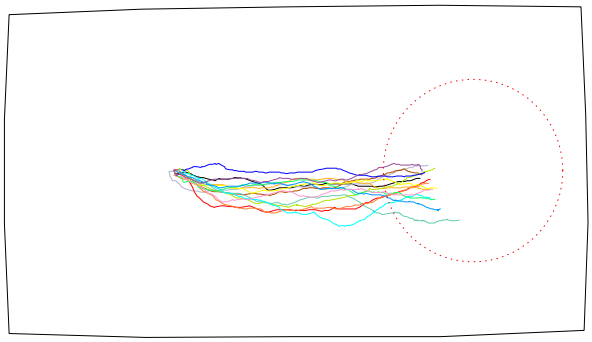


(b) Triangular Object

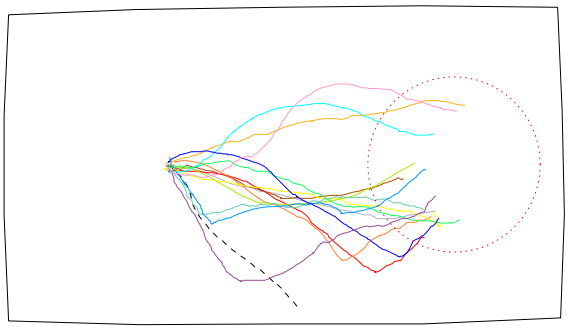


(c) Rectangular Object

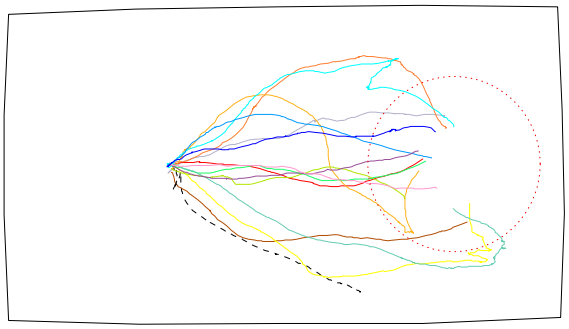
Figure 4.9: Predicted paths of the centroid of the objects based on Lemmas 1 and 2. These paths are plotted using the same ratio on both of the axes, so they can be compared with Fig. 4.10. The length unit of the x and y axis in these plots are centimeter.



(a) Circular Object



(b) Triangular Object



(c) Rectangular Object

Figure 4.10: Actual paths of the centroid of the objects. The dashed black lines are the paths of the two failed trials. The dotted red line is the goal region. It can be observed that the strategy has an effect to correct the direction in which the object is moved. Sometimes, this correction resulted in a significant change in the transport direction.

object, it is possible to predict the objects' paths for the trials. The predicted paths are given in Fig. 4.9. In addition, the actual paths of the objects were traced from the videos recorded by the overhead camera. These true paths are given in Fig. 4.10.

The differences between each pair of individual paths in Figs. 4.9 and 4.10 are obvious; in only some trials, the prediction is close to the actual paths. This result however was expected since many of the idealized assumptions made in Section 4.3 are violated in a physical environment. For example, the robots will not be able to react instantaneously to changes in the object's occluded perimeter. Moreover, the robot's embodiment raises the issue of physical interferences. However, the overall distributions of the paths show a good correspondence:

- The circular object tends to move directly to the goal.
- The paths of the triangular object are typically curved. This object is difficult to be pushed along a straight line towards the goal.
- The paths of the rectangular object have a more random but uniform distribution.

Path Efficiency

We define the path efficiency of a trial as:

$$PE = \frac{s_{min}}{s}. \tag{4.26}$$

s_{min} is the length of the shortest straight line from the start position to the goal region. s is the length of the path of the object when its centroid enters the goal region. An ideal transport path should have a PE of 1.0.

For all successful trials, both the actual PE values and the PE values corresponding to the predicted paths shown in Fig. 4.9 are calculated. Fig. 4.11 shows a box-and-whisker plot of predicted PE versus actual PE for each of the objects. The predicted and actual PEs of an object both indicate the difference in the efficiency when transporting objects of different shapes.

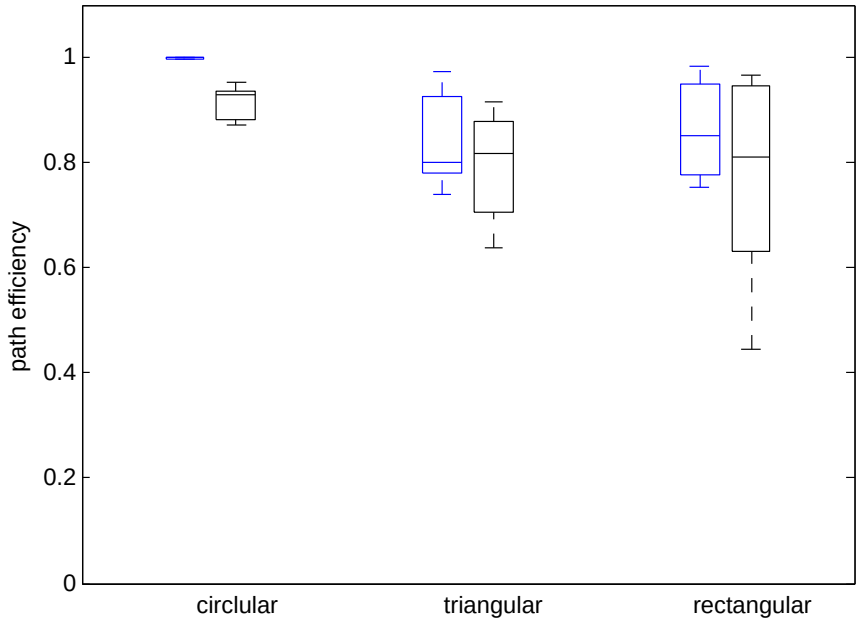


Figure 4.11: Path efficiency in the successful trials. This metric compares the length of the path that the object moved against the length of the ideal straight path to reach the goal. For each of the objects, the predicted PE and actual PE are blue (left) and black (right) respectively.

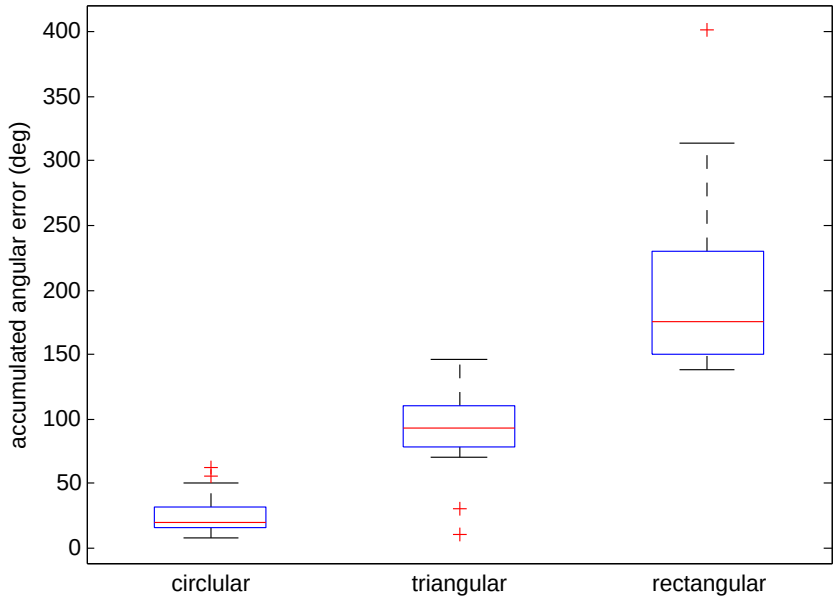


Figure 4.12: Accumulated angular error when the object enters the goal region. This metric reflects how much unnecessary rotation appeared in the transportation.

Accumulated Angular Error (AE)

The efficiency of a pushing-based transport strategy may also be affected if a substantial amount of unnecessary object rotation occurs in the process.

We define the accumulated angular error as the difference between the relative difference in the orientations at the beginning and the end of a trial and the total amount of changes of orientation. Let $\mathbf{p}(t)$ and $\mathbf{q}(t)$ be the centroids of the two tracking markers on top of the object in the video of a trial at time step t. Then, the orientation vector of the object at time step t is

$$\mathbf{a}(t) = \mathbf{p}(t) - \mathbf{q}(t). \tag{4.27}$$

The step interval used in the off-line video tracking is 1 s. The change of the orientation between two time steps, t_0 and t_1, is defined as:

$$D(t_0, t_1) = \left| \arccos \frac{\mathbf{a}(t_0) \cdot \mathbf{a}(t_1)}{\|\mathbf{a}(t_0)\| \, \|\mathbf{a}(t_1)\|} \right|. \tag{4.28}$$

The accumulated angular error is calculated as:

$$AE = \left| D(T_k, 0) - \sum_{t=1}^{T_k} D(t, t-1) \right|. \tag{4.29}$$

Note, the relative difference between the object's initial orientation and its orientation when it reaches the goal ($D(T_k, 0)$) is excluded, because we focus on quantifying the wasted effort on rotation. e.g. two continuous rotations that cancel each other.

This metric will be zero if the transport process is ideal. Fig. 4.12 shows the box-and-whisker plot of the accumulated angular error of the successful trials. Due to the length of the elongated rectangular object, randomness in the distribution of the pushing robots can cause a torque that is big enough to rotate the object rapidly. However, it is also due to the randomness in such rotations that this object will not always point with one of its ends towards the goal, which would cause the occluded surface for pushing to be very small.

4.6 Experiments with A Moving Goal

In a more complex environment, the goal may not be perceived from any position around the object. For example, there could be obstacles between the object and

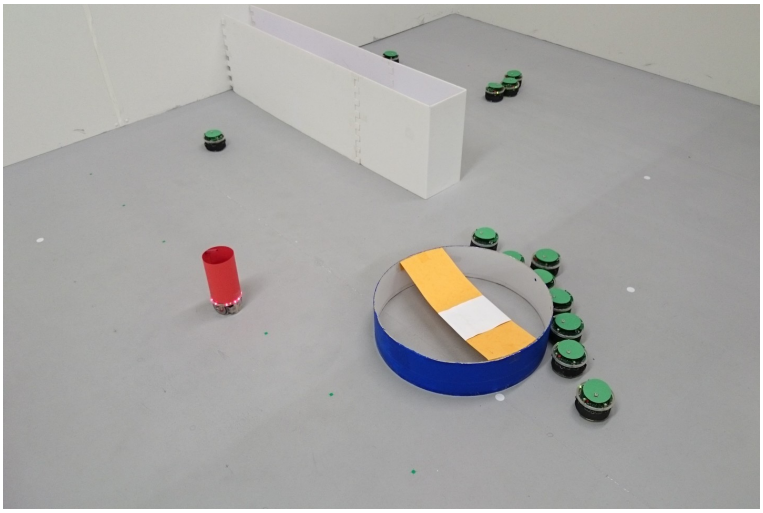


Figure 4.13: In this transport strategy, the position of the goal can be changed during the transport process. Therefore, a mobile robot can be used as the goal. In this photo, the group of e-pucks are pushing the object towards the e-puck with a red column on its top (the goal robot). The goal robot can be moved at any time via Bluetooth remote control.

the goal, or the distance between them could be bigger than the range of sensors of the robots. The transport strategy as it stands can not deal with such an environment. However, it is possible to adapt the goal in the strategy to expand the capability of the transport system.

If the goal is a mobile robot, it can change its position when the object arrives. If this goal robot has enough intelligence, it can change its position along a route that guides the transport robots to the final destination. How to implement such an intelligent goal robot is a research topic in itself [5] [89]. In this section, we present an experiment in which a tele-operated goal robot was used to guide the pushing robots through a corridor with corners.

4.6.1 Implementation

The e-pucks in charge of pushing the object (the transport robots) used the controller exactly as introduced in the previous section. In other words, these e-pucks are programmed to push a blue object to a red goal.

An extra e-puck was used to implement a mobile goal (the goal robot). To make this robot be perceived as the goal, a red cylinder was placed over it (see

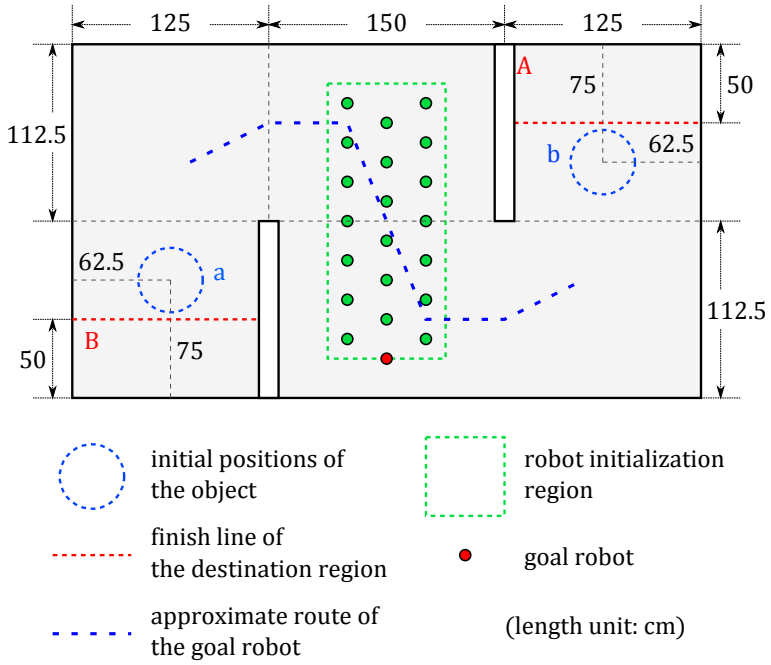


Figure 4.14: Setup for experiments with a moving goal. The initial position of the object was alternated between a and b while their corresponding destination regions were A and B.

Fig. 4.13). To further increase its visibility, it kept all of its red LEDs turned on.

The goal robot was driven remotely by a human operator via Bluetooth. Because the transport robots push the object towards the goal robot, the operator can indirectly control the transport direction by driving the goal robot.

4.6.2 Experimental Setup

Environment

The experiment was setup in the same arena as the one in the previous section while using the same circular object. Fig. 4.14 illustrates the setup. Two walls were placed within the environment. The initial position of the object was alternated between the bottom left corner and the top right corner of the arena. The destination was a rectangular region opposite the initial position of the object. The direct line of sight between the object's start position and the destination were blocked by the walls.

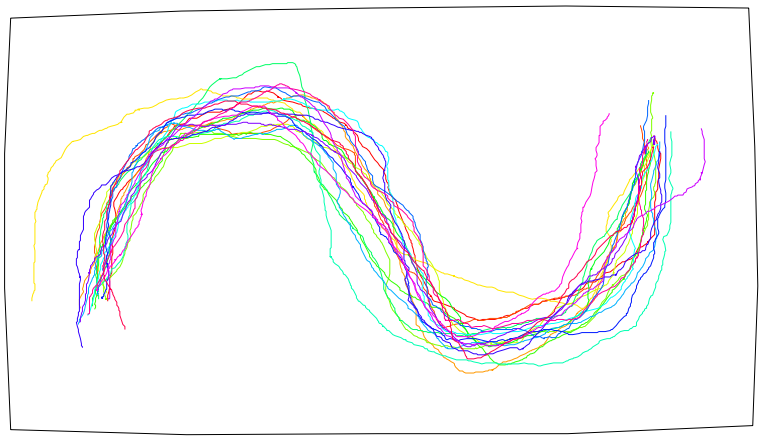


Figure 4.15: The traces of object's centroid.

Trial Procedure

The human operator was required to move the guiding robot along a designated path. The path was specified by a series of way points (see Fig. 4.14). When the distance between the object and the goal robot was very small, the operator moved the goal robot to the next way point. When the object touched the destination region (finish line), the trial was considered successful.

4.6.3 Results

In all 20 trials, the object reached the destination region. The mean and median of the completion times are 859 sec and 861 sec respectively. The minimum and maximum are 649 sec and 1086 sec respectively.

The traces of the object's centroid is shown in Fig. 4.15. From the plot, it is clear that the object generally followed the designated route of the goal robot.

According to these results, the transport strategy is able to deal with a moving goal. This means the transport strategy can potentially become part of a more complex behavior to autonomously complete transport tasks in a more complex environment. From another point of view, the human operator successfully commanded the swarm of robots to achieve an object transportation task through remote control.

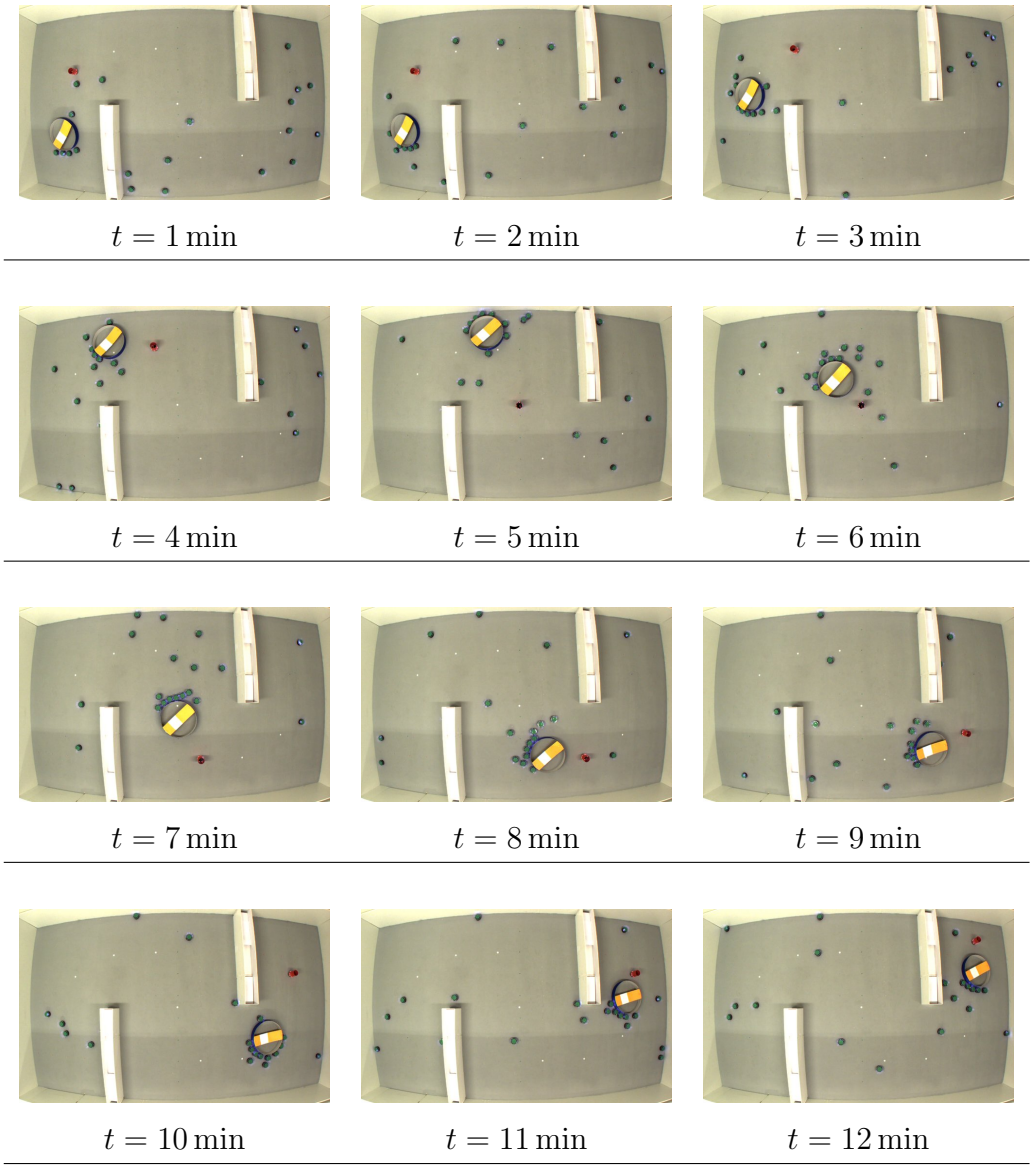


Figure 4.16: Snapshots of one of the trials in the systematic experiments where the transport group pushes the object towards a tele-operated goal robot and thereby through an environment with obstacles.

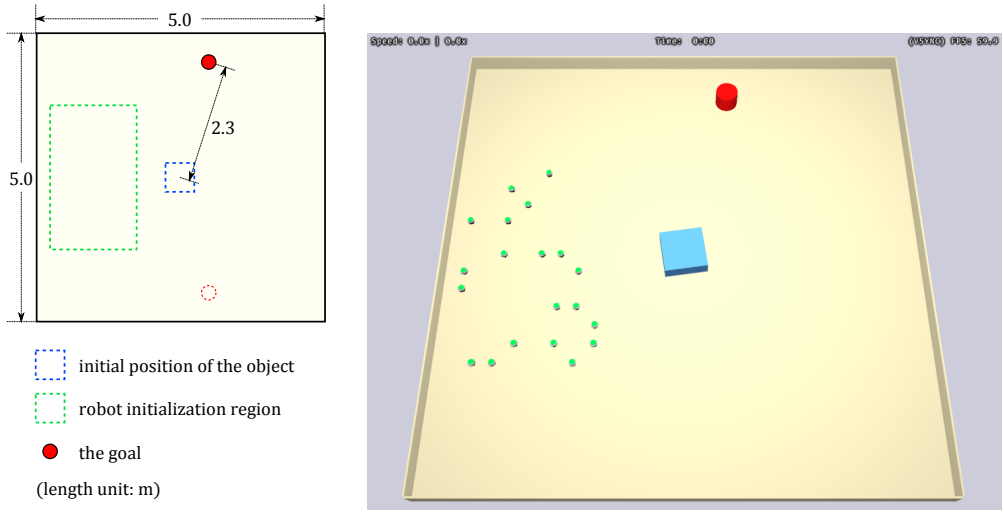


Figure 4.17: Setup of the simulation evaluating the scalability of the transport strategy.

4.7 Simulations Varying Number of Robots

To evaluate the scalability of the transport strategy, the number of robots in the group needs to be varied in a wide range. This is difficult to do in real environment due to the number of e-pucks available and the time cost to operate a large number of e-pucks. Therefore, a computer simulation was established to study this problem in which the number of robots (group size) was varied from 10 to 80 with 10 as incremental step.

When using the simulation framework introduced in Chapter 3, the simulated model of the e-puck is relatively realistic. For example, the proximity sensors on the simulated e-puck output values that are similar to those obtained on a real e-puck. Therefore, the controller for the simulated e-puck can be nearly identical to the one used on the real e-puck.

4.7.1 Simulation Setup

The environment was a bounded arena of size 500 cm × 500 cm.

The object was a cuboid box. Its side length was varied from 20 cm to 60 cm with 10 cm as incremental step. Its mass (in gram) was 0.4 times the square of its side length (in cm), so the mass of an object with side length of 20 cm would be

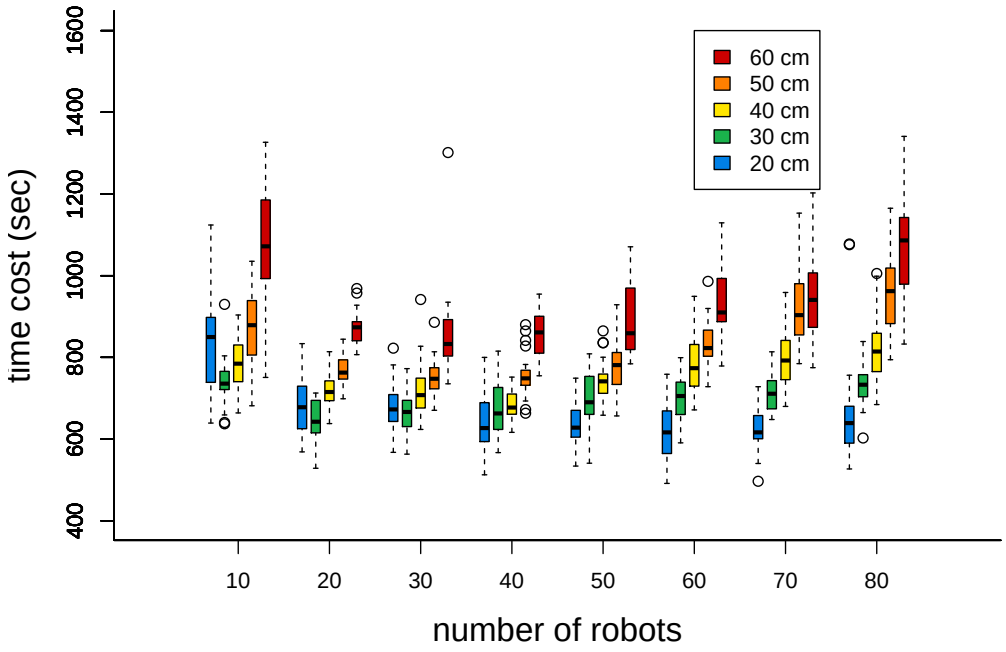


Figure 4.18: The box-plot showing completion time of the trials. Each box/pillar in the plot represent 30 trails.

160 gram.

The overall configuration is illustrated in Fig. 4.17. In the beginning of a trial, the robots were randomly distributed in a rectangular region of size 2.4 m × 1.6 m. The initial position of the object was in the center of the arena. The goal was placed 2.3 m from the center of the object. The relative orientation between the goal and the object was randomly chosen from 75 deg or −75 deg.

For each size of the object and each group size, 30 trials were executed. When the distance between the object and the goal was smaller than 50 cm, the trial would be considered successful and stopped. The time limit of a trial was 1200 seconds of simulation time. The trial would be stopped and considered as failed once the time limit was reached.

4.7.2 Results

In all of the trials, the group of robots successfully pushed the object to the goal. Videos of typical situations in these trials can be found in [25].

Fig. 4.18 shows the completion time of the trials. It can be seen that an optimal

group size of about 30 exists commonly for objects of all sizes. In the setup, the cuboid object had a maximum side length of 60 cm, which implies 30 robots of diameter 7 cm is enough to cover two of its sides. To further increase the number of robots may reduce the efficiency slightly. Such a result is consistent with other works, e.g. Vaughan [134]. According to the observation of the trials, robots in front of the object may collide with the object temporarily when it is moving. The more robots that are not in the pushing formation, the more likely there will be robots in front of the object and thereby slow down the process. However, as long as the environment is not overly crowded with robots, such interferences should not cause the transport to fail because the force exerted by individual robot are insignificant when compared to the combined pushing force of the robot in the pushing formation.

4.8 Simulations in 3-D Environment

The transport strategy has potential to be implemented in a 3-D environment. In this section, we present a conceptual implementation of the occlusion-based transport strategy in a simulated 3-D environment[3]. The environment is a bounded gravity-less rectangular space. The speed of any objects in this space are damped such that consistent forces are required to maintain the motion of objects. These conditions approximate underwater environments where the density of the object equals the density of water. One hundred robots were deployed in this environment to push an object towards a goal. The goal was set to be the dominant light source in the environment. The robots were required to push across the portion of the object's surface where the direct light from the goal was occluded by the object. Fig. 4.19 shows this scenario.

4.8.1 Conceptual Robot Design

A specific robot model was designed for the task (see Fig. 4.20). Following the concept of swarm robotics, the capability of the robot was kept simple. The robot is modeled as a cylinder of diameter 8 cm and height 6 cm. Its mass is 300 g. It is

[3]This simulation did not used the framework introduced in Chapter 3.2. It is based on Bullet Physics library alone, which is used to handle rigid body physics.

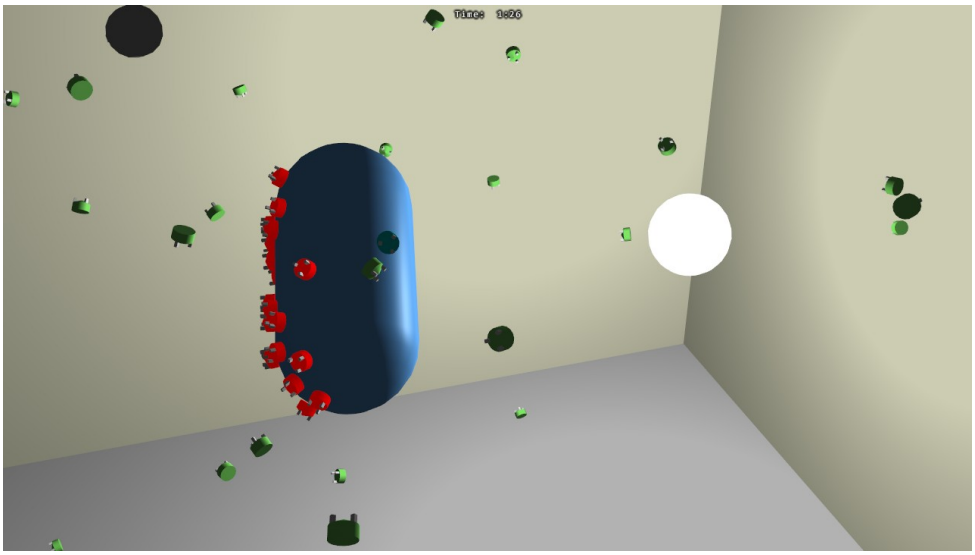


Figure 4.19: In this 3-D physics-based simulation, a swarm of robots are pushing an object (the blue capsule) towards a light source (the white sphere). The robots only push across the shadow side of the object where the direct line of sight to the goal light is occluded by the object.

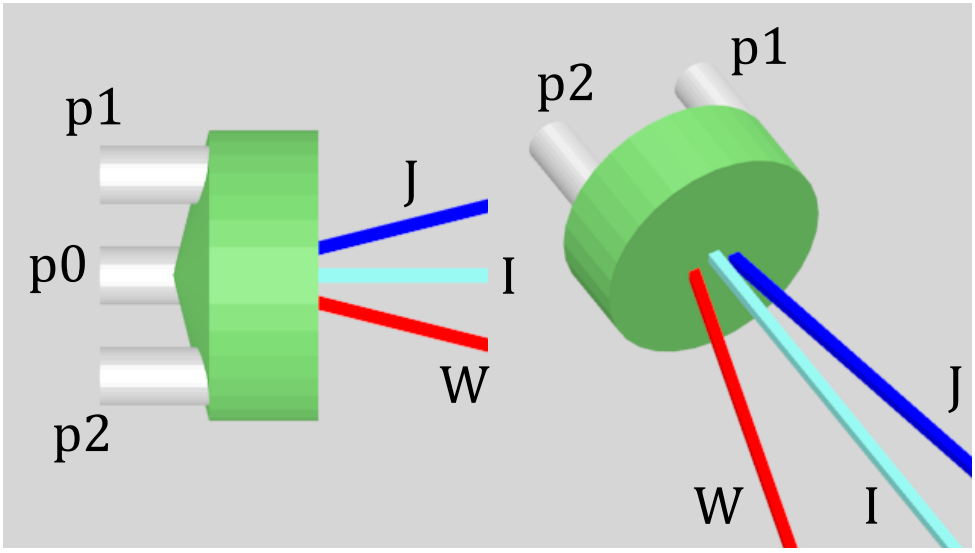


Figure 4.20: The graphic model of the conceptual robot used in the simulation of the transport task in a 3-D environment. In this image, the three engines of the robot and the beams of the three line of sight sensors are rendered in detail.

propelled by three engines mounted on its back side. Each of them can generate a thrust force both forward or backward, denoted as p_0, p_1 and p_2.

The robot has four sensors that give binary outputs:

1. I: Long range object sensor. This sensor can detect whether there are objects along the line of sight of it. Its normal vector (pointing direction) is $(1.0, 0.0, 0.0)$ in the robot's local coordinate system. Its range is 1000 cm.

2. J: Short range object sensor. This sensor can detect whether there are objects along the line of sight of it. Its normal vector is $(1.0, 0.57, -0.57)$ in the robot's local coordinate system. Its range is 40 cm.

3. K: Ambient light sensor. This sensor can detect whether the robot is directly illuminated by the goal light source. It simply checks the line of sight between the robot and the goal light.

4. W: Obstacle sensor. This sensor can detect whether there are obstacles along the line of sight of it. Walls, other robots and the goal light are considered as obstacles in the environment. The sensor's normal vector is $(1.0, -0.57, -0.57)$ in the robot's local coordinate system. Its range is 40 cm.

4.8.2 Robot Controller

The thrust outputs of the three engines are computed from three motion outputs – speed, yaw and pitch – as follows:

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 1, 0, -1 \\ 1, -1, 0.5 \\ 1, 1, 0.5 \end{bmatrix} \begin{bmatrix} speed \\ yaw \\ pitch \end{bmatrix}. \tag{4.30}$$

The robot has four binary sensor inputs. For each sensor value combination, a set of motion outputs was defined to achieve the desired motion in the transport strategy. The mapping is given in Table 4.4. Ranges indicate that the value was uniformly random chosen from with the range.

4.8.3 Simulation Setup

One hundred robots were randomly placed in a rectangular space of dimension 800 cm × 500 cm × 500 cm.

Table 4.4: Mapping from inputs to outputs

W	I	J	K	*speed*	*yaw*	*pitch*
0	0	0	–	0.6	$[-0.03, 0.07]$	$[-0.1, 0.1]$
0	0	1	0	0.2	$[-0.03, 0.17]$	$[-0.1, 0.1]$
0	0	1	1	0.3	$[0.02, 0.12]$	$[-0.3, 0.3]$
0	1	0	–	0.8	0.0	0.0
0	1	1	0	0.7	$[-0.2, 0.2]$	$[-0.2, 0.2]$
0	1	1	1	0.0	$[-0.13, -0.03]$	$[-0.1, 0.1]$
1	–	–	–	-0.8	$[-0.3, 0.3]$	$[-0.3, 0.3]$

Consider the environment as a box of which the two diagonal vertices have coordinates of $(0, 0, 0)$ and $(800, 500, 500)$ in the global coordinate system. The goal light was fixed at position $(650, 250, 250)$. The object was initialized at $(280, 250, 250)$ while its initial orientation was randomized using uniform spherical distribution.

Four types of objects were used:

1. a sphere with a radius of 41 cm;
2. a capsule with a side length 60 cm and a radius of 30 cm;
3. a cube with a side length 66 cm;
4. a cone with a height of 100 cm and a radius of 52 cm.

The mass of these objects were all approximately 280 kg (calculated from their volumes using the density of water).

For each type of object, 100 simulation trials were run. When the centroid distance between the object and goal light was less than 90 cm, the trial would be stopped, and considered successful. The trial would also be stopped when 900 sec elapsed. This time limit was chosen to be about 5 times the typical finishing time in preliminary trials.

4.8.4 Simulation Results

In all 400 trials, the object reached the goal within the time limit. The box plot of the completion times (in simulated seconds) for each of the objects is shown in Fig. 4.21. The path efficiency of the trials are shown in Fig. 4.22.

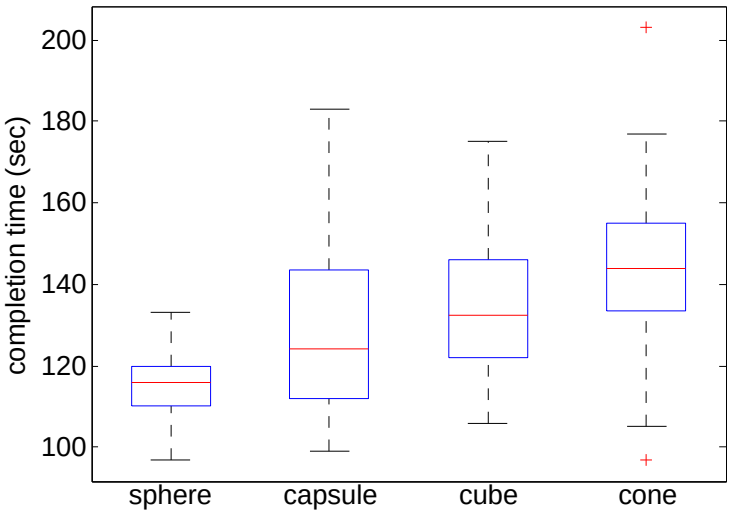


Figure 4.21: Compeletion time of the simulation trials in a 3-D environment.

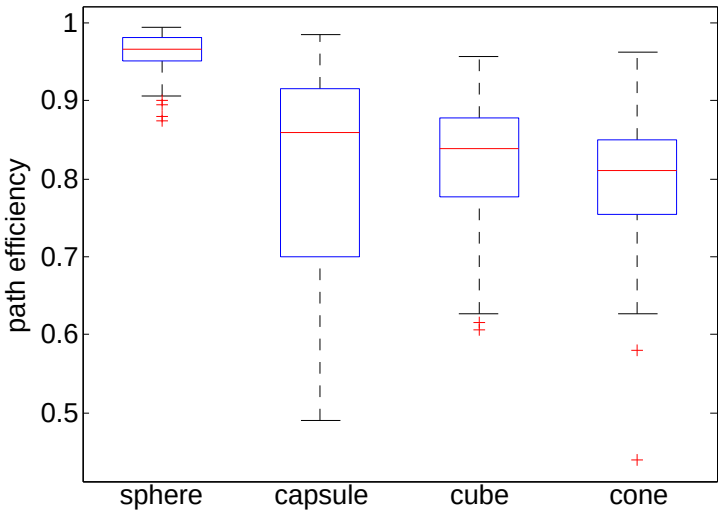


Figure 4.22: Path efficiency of the simulation trials in a 3-D environment.

Typical situations of the four objects are shown in the online supplementary material [25].

According to both of the numeric results and the direct observation, the transport task was successfully completed by the robots. Similar to the 2-D case, the shape of the object affects the performance of the strategy.

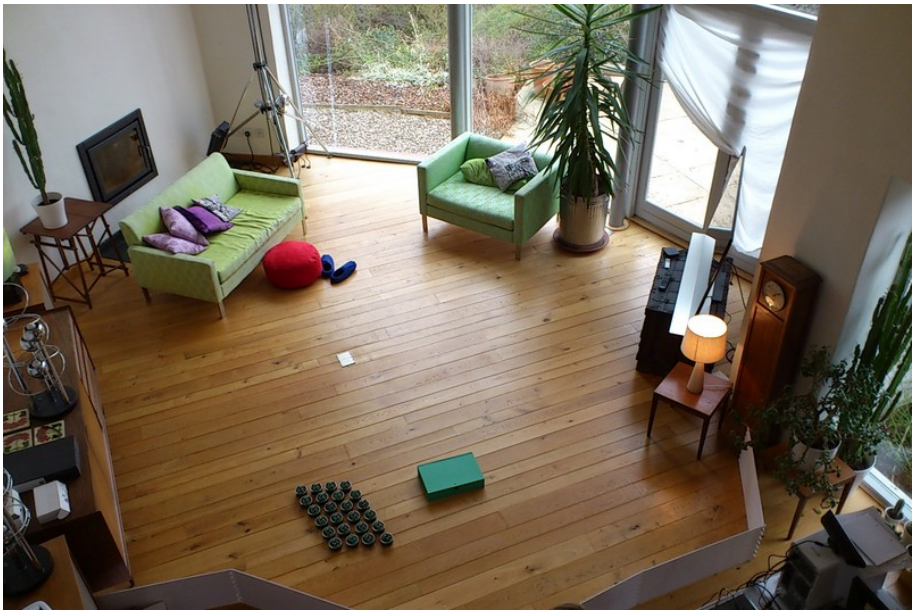


Figure 4.23: The cooperative transport system studied in this thesis has been used for a demonstration in a domestic environment.

4.9 Demonstration in a Domestic Environment

The cooperative transport system implemented on the e-puck was also used to demonstrate a cooperate object pushing task in a domestic environment for a TV broadcast on Channel 5[4]. In the scenario showing in Fig. 4.23, a group of e-pucks were required to push a pair of blue slippers (the object) to a red cushion seat (the goal). There was no major obstacle between the object and the goal. Therefore, the transport strategy could be applied directly.

In such a domestic environment, the ambient infrared intensity from the daylight is strong. Therefore, the working mode of the infrared proximity sensors was changed to "EMITTER NOISE" (see Section 3.1.3 for details). To cope with this change, the weighting parameters used in the motion controller were also re-tuned. Moreover, the e-puck's hardware design was just robust enough to cope with the unevenness of the wooden floor. Occasionally, a robot would change direction when its wheel interfered with the interface between the floor boards.

[4]for details, check: `http://naturalrobotics.group.shef.ac.uk/`

Chapter 5

Segregation Based on Brazil Nut Effect

When a mixture of particles with different attributes undergoes vibration, a segregation pattern is often observed. For example, in muesli cereal packs, the largest particles—the Brazil nuts—tend to end up at the top. For this reason, this phenomenon is known as the Brazil nut effect. In previous research [51], a decentralized robotic controller inspired by this phenomenon was designed to produce segregation patterns in swarms of simulated agents that move on a plane.

In the work introduced in this chapter, this controller was implemented on real e-puck robots. In a swarm of e-pucks, different robots mimic disks of different sizes (larger than their physical dimensions). The motion of every robot is governed by a combination of three components: (i) attraction towards a common reference point, which emulates the effect of a gravitational pull, (ii) random motion, which emulates the effect of vibration, and (iii) repulsion from nearby robots, which emulates the effect of collisions between disks. The algorithm does not require robots to discriminate between other robots; yet, it is capable of forming annular structures where the robots in each annulus represent disks of identical size. The controller was verified through systematic experiments performed with a group of physical e-pucks.

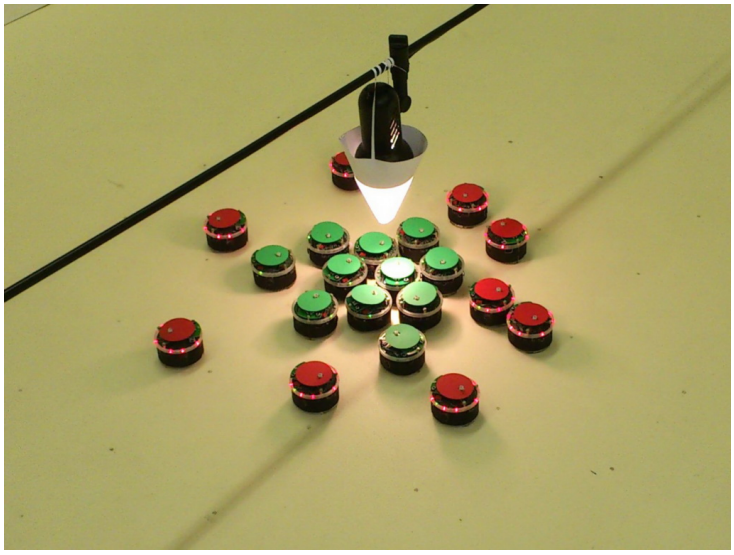


Figure 5.1: An annular segregation pattern in a swarm of 20 e-puck robots.

5.1 Objectives

Segregation is a process whereby objects or individuals separate into distinct groups. It can be observed on various scales, ranging from the molecular to the macroscopic scale.

The Brazil nut effect [101] refers to the segregation that occurs when shaking a mixture of granular material of different sizes. When shaken, the packing of the particles will loosen. Therefore, there is a probability for the smaller particles to move into the void between the larger particles. However, it is less probable for larger particles to move into the void between the smaller particles. Such a unidirectional process causes the overall distribution of the particles to become segregated according to their size [8].

The Brazil nut effect can be utilized to create a spatial segregation among a swarm of robots in a decentralized manner. In [51], a segregation algorithm based on the Brazil nut effect was developed and tested in computer simulation. Through emulating the Brazil nut effect with different particle sizes in a 2D space, those homogeneous robots should be able to form a pattern that shows a spatial segregation based on size of the particles they are mimicking. This chapter aims to implement and verify this algorithm on a swarm of e-pucks in the real world.

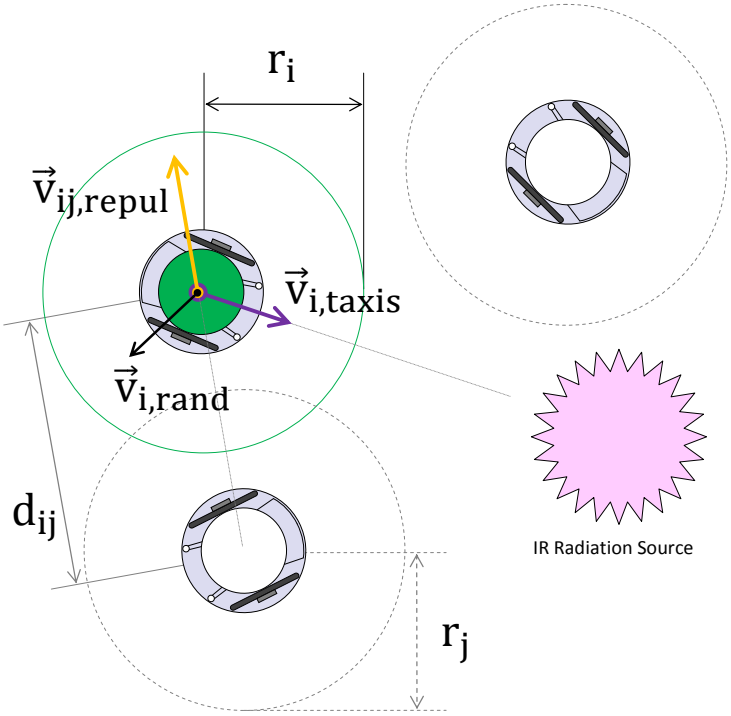


Figure 5.2: The three behavioral components of robot i. Vector $\vec{v}_{i,\text{taxis}}$ points towards the estimated location of the infrared radiation source. Vector $\vec{v}_{i,\text{rand}}$ points in a random direction. Vector $\vec{v}_{ij,\text{repul}}$ is due to the repulsion effect on robot i by robot j. Robot i is repelled by robot j if it perceives the virtual body of robot j as intersecting with its own virtual body. As robot i has no means of measuring the virtual radius of robot j (r_j), it assumes that $r_j = r_i$.

5.2 Controller Implementation

The controller is based on the algorithm presented in [51]. Some assumptions in the algorithm cannot be realized for the physical e-puck. Therefore, modifications had to be applied to the algorithm. This section will describe the algorithm used here and highlight the modifications made.

5.2.1 Motion Composition

The motion controller is implemented based on the motor schema paradigm [3], in which the velocity of the robot is calculated from a weight summation of a number of vectors.

The robots emulate a mixture of differently-sized disks subjected to vibration on a 2-dimensional plane. In particular, robot i emulates a disk of radius r_i, whose motion is governed by a combination of three components (see Fig. 5.2):

1. $\vec{v}_{i,\text{taxis}}$: attraction towards a point common to all the disks, which emulates the effect of a gravitational pull,

2. $\vec{v}_{i,\text{rand}}$: random motion, which emulates the effect of vibration and

3. $\vec{v}_{i,\text{repul}}$: repulsion from nearby disks, which emulates the effect of collisions.

Hereafter, the disk a robot represents is also referred to as the *virtual body* of the robot. The radius of the disk is also referred to as the *virtual radius* of the robot.

In every control cycle, robot i calculates the aforementioned three vectors. These are then combined as follows:

$$\vec{v}_i = \vec{v}_{i,\text{taxis}} + c_{\text{rand}}\vec{v}_{i,\text{rand}} + f(\vec{v}_{i,\text{repul}}). \tag{5.1}$$

Vector $\vec{v}_{i,\text{taxis}}$ is always a unit vector. Vector $\vec{v}_{i,\text{rand}}$ is also a unit vector but a parameter c_{rand} is used to weight its magnitude. Vector $\vec{v}_{i,\text{repul}}$ can have a large magnitude because it is computed as a sum of possibly many vectors (for details, see Section 5.2.1); therefore, its magnitude is capped by function $f(\cdot)$. Here, we use $c_{\text{rand}} = 0.6$ and a maximum allowed magnitude of 6.4 units for $\vec{v}_{i,\text{repul}}$. These settings follow suggestions from simulation results[1] [51].

After constructing motion vector $\vec{v}_i$, robot i first turns to point in its direction, and then moves forward for a fixed duration. The speed at which it moves forward is proportional to the magnitude of the vector, so that the maximum magnitude possible (i.e.,$1 + 0.6 + 6.4 = 8$ units) corresponds to the maximum speed of the robot (12.8 cm/s).

The length of the control cycle used here is 5 s, which is substantially longer than that used in simulation (0.1 s). The main reason for this is that the e-puck robots are equipped with directional cameras, whereas the simulated robots had omni-directional perception [51]. In each cycle, the robot spends around 2.4 s in

[1]The algorithm in [51] uses an additional parameter to weight $\vec{v}_{i,\text{repul}}$. This is not used here because the repulsion mechanism has been modified. The weightings used here are identical to [51] when one considers the maximum allowed magnitude of $\vec{v}_{i,\text{repul}}$.

revolving to obtain an omni-directional image, 1.3 s in turning to point in the direction of $\vec{v}_i$, and 1.3 s in moving forward.

The implementation is based on the sensing framework introduced in Chapter 3. Thus, the e-puck's infrared proximity sensors and camera are the sensor inputs. In the following, we detail how vectors $\vec{v}_{i,\text{taxis}}$, $\vec{v}_{i,\text{rand}}$ and $\vec{v}_{i,\text{repul}}$ are computed based on the sensor inputs.

Attraction to Center of Gravity

The algorithm requires a point of attraction in the environment to emulate the effect of a gravitational pull. Each robot is required to estimate the angular position of this point (the distance to it is not needed).

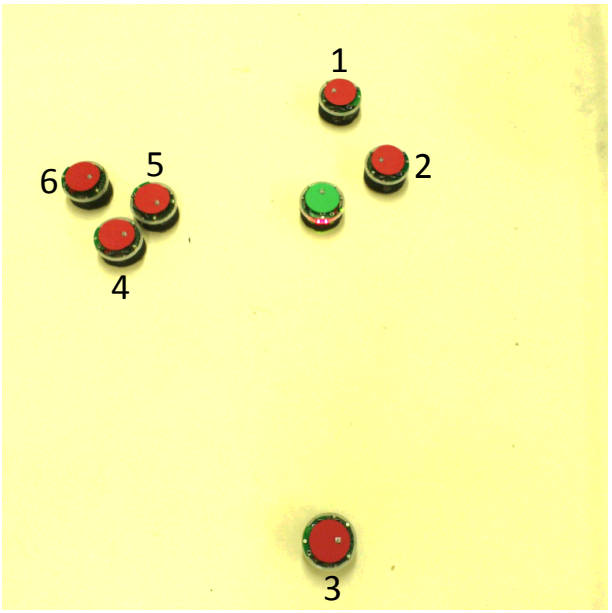
In our experimental setup, we use an infrared radiation source—a light bulb—as the point of attraction. In order to estimate its angular position, each robot makes use of its eight infrared sensors. In every control cycle, the three sensors giving the highest readings are selected. Each reading is then represented as a vector pointing from the center of the robot to the physical location of the sensor, with a magnitude proportional to the sensor's reading. The three vectors are summed, and the resulting vector is normalized to have a unit magnitude, giving $\vec{v}_{i,\text{taxis}}$.

Random Motion

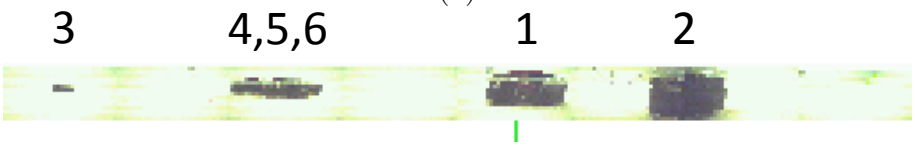
The random motion vector $\vec{v}_{i,\text{rand}}$ is taken to be a unit vector pointing in a random direction in the interval $[0, 2\pi)$. This direction is taken with respect to the robot's orientation at the beginning of the control cycle.

Repulsion

In principle, each robot should be repelled by every other robot whose virtual body overlaps with its own virtual body. This would require the robots to know the virtual radii of nearby robots. However, as shown in [51], segregation can still be effectively achieved if every robot assumes for all other robots a constant virtual radius, which is a parameter that needs to be fixed a prior. Here, we propose and use an alternative, parameter-free heuristic: robot i assumes that the virtual radius of all other robots is equal to its own, that is, r_i.



(a)

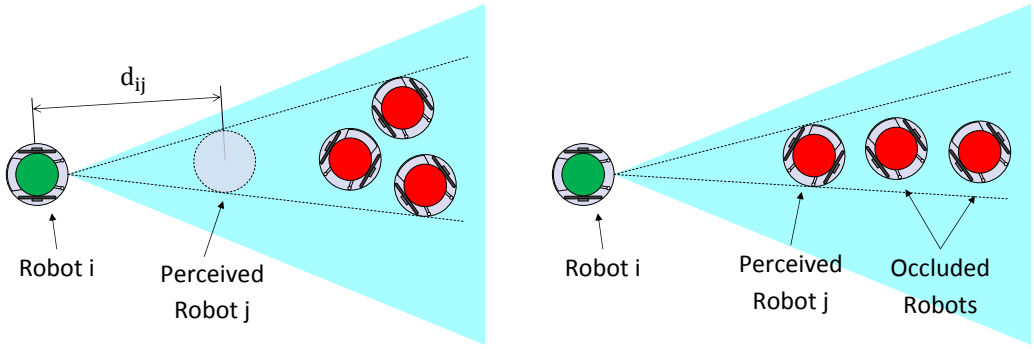


(b)

Figure 5.3: Image processing to analyze the distances to nearby robots. (a) Overview of a scene with seven robots. (b) The corresponding concatenated image (here, with the original 15 pixel height) formed by the green robot as it takes eight images in one revolution. Note how the green robot sees the red robots 4, 5 and 6 as a single object that appears closer (see also Fig. 5.4).

5.2.2 Distance Sensing

Each of the e-pucks uses its camera to estimate the angular position of and distance to nearby robots. In every control cycle, a robot turns through one revolution in eight steps of 45° each. In each step, its camera takes a picture. From the center of this picture, a horizontal line of 32 pixels is extracted (corresponding to a field of view of 45°). The pixel lines extracted from the eight images are concatenated to give a panoramic view of the scene (see Fig. 5.3). The concatenated image is traversed horizontally to scan for nearby robots. This is achieved by identifying blocks of dark pixels. Each block represents a perceived robot j. The angular position of that robot is estimated from the position of the block. Vector $\vec{v}_{ij,\text{repul}}$



(a) (b)

Figure 5.4: Possible misperceptions. (a) Robot i sees three overlapping robots as a single object, j. It incorrectly perceives a single robot at distance d_{ij}. (b) Robot i can not see the two robots occluded by robot j.

points in the direction away from robot j. The distance to the robot, d_{ij}, is estimated from the width of the block. The amount of repulsion from a perceived robot j is proportional to the perceived amount of intersection. Formally,

$$||\vec{v}_{ij,\text{repul}}|| = \begin{cases} k\,(2r_i - d_{ij}) & d_{ij} < 2r_i; \\ 0 & d_{ij} \geq 2r_i, \end{cases} \tag{5.2}$$

where $k = 0.2$.

The total repulsion on robot i, $\vec{v}_{i,\text{repul}}$, is giving by summing the individual repulsion vectors for all blocks.

The vision based implementation differs from [51] in that two types of misperceptions can occur: (i) it is possible for several robots to be perceived as a single block of pixels (see Fig. 5.4(a)); (ii) it is possible for a robot to occlude one or more robots completely (see Fig. 5.4(b)). In order to compensate for these misperceptions, the repulsion mechanism places more emphasis on robots that are perceived to be close (see (5.2)). This is in contrast with the mechanism used in simulation [51], where the amount of repulsion is constant regardless of the distance to a perceived robot.

5.3 Experiments using Two Groups of Robots

5.3.1 Experimental Setup

We use n to denote the number of robots in the swarm. Furthermore, we use m to denote the number of groups, and n_k to denote the number of robots in group k, $k \in \{1, 2, \ldots, m\}$. The robots in group k all have virtual radius $r^{(k)}$. Recall that r_i denotes the virtual radius of robot i. Thus, $r_i = r^{(k)}$, if robot i is in group k.

We consider a system with $n = 20$ robots and with $m = 2$ different groups. The virtual radius of robots from group k is chosen as follows:

$$r^{(k)} = ab^{k-1}, \tag{5.3}$$

where a is the size (in cm) of the smallest disk and b is the minimum size ratio between disks of different groups. We use $a = 8\,\text{cm}$ and $b \in \{1, 2, 3, 4, 5\}$.

Ideally, we expect the robots to organize into an annular structure, where the disks of radius $r^{(k)}$, $k \in \{1, 2, \ldots, m\}$, are fully contained within the area of the annulus formed by the concentric circles of radii $(k-1)g$ and kg in the center of the environment. Parameter g represents the "thickness" of the annulus and can be controlled by group size n [51].

An approximation of the ideal pattern can be obtained by choosing n_k as follows [51]:

$$n_k = \frac{\frac{2k-1}{(r^{(k)})^2}}{\sum_{j=1}^{m} \frac{2j-1}{(r^{(j)})^2}} n. \tag{5.4}$$

In our physical implementation, the robots moved in a square arena of sides 2.5 m. A light bulb was placed over the center of the arena, acting as the infrared radiation source, that is, the point of attraction.

The initial placement of the robots was done as follows: a square grid of 6×6 points was marked on the arena floor, centered around the light bulb, with all points being 20 cm apart. For each trial, 20 points were chosen randomly without replacement. Additionally, for each robot, the orientation was selected randomly from four possibilities: north, south, east and west.

Each trial was recorded from start to finish with an overhead camera system. All robots were started simultaneously using an infrared remote control.

They would stop automatically after 20 minute elapsed (duration of a trial was 20 minute).

5.3.2 Performance Metric

To measure the quality of segregation, we calculate the *segregation error* (SE) as defined in [51]. Consider two robots i and j and let $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ denote their positions.

Furthermore, let $\boldsymbol{o}$ denote the position of the 'center of gravity' in the same co-ordinate system, that is, the point to which all robots are attracted.

The pair of robots (i, j) contributes to the segregation error if one of the robots has a larger virtual radius *and* is closer to $\boldsymbol{o}$ than the other one. It does not contribute to the segregation error if either the robots have identical virtual radii, or if the robot with a smaller virtual radius is closer to $\boldsymbol{o}$ than the other one. Formally,

$$e_{ij} = \begin{cases} 1 & (r_i < r_j) \wedge (\|\boldsymbol{x}_i - \boldsymbol{o}\| \geq \|\boldsymbol{x}_j - \boldsymbol{o}\|)\,; \\ 1 & (r_i > r_j) \wedge (\|\boldsymbol{x}_i - \boldsymbol{o}\| \leq \|\boldsymbol{x}_j - \boldsymbol{o}\|)\,; \\ 0 & \text{otherwise.} \end{cases}$$

The segregation error is given by summing e_{ij} over all pairs of robots, and normalizing by (only) the number of errors possible. Formally,

$$\mathrm{SE} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij}}{n^2 - \sum_{k=1}^{m} n_k^2}, \tag{5.5}$$

where $\mathrm{SE} \in [0, 1]$. Randomly placed robots will have a segregation error of 0.5 on average. An error of 1.0 is achieved if the robots are in an 'inverse Brazil nut' configuration, that is, if for all (i, j) s.t. $r_i < r_j$, $\|\boldsymbol{x}_i - \boldsymbol{o}\| \geq \|\boldsymbol{x}_j - \boldsymbol{o}\|$.

We considered $m = 2$ groups of robots. Robots of the first group represented disks of radius $r_1 = 8\,\mathrm{cm}$, whereas robots of the second group represented disks of radius $r_2 = 8b\,\mathrm{cm}, b \in \{1, 2, 3, 4, 5\}$. As reported in [146], the size ratio b is a critical variable—increasing it results in a decrease in the segregation error.

For each value of b, we performed 20 trials with $n = 20$ robots each, that is, we ran 100 experimental trials in total. Every trial lasted for 20 minutes. Table 5.1 shows the number of robots in each group see (5.4).

Table 5.1: Overview of configurations studied

radius factor b	n_1	$r^{(1)}$	n_2	$r^{(2)}$
1.0	5	8.0 cm	15	8.0 cm
2.0	11	8.0 cm	9	16.0 cm
3.0	15	8.0 cm	5	24.0 cm
4.0	17	8.0 cm	3	32.0 cm
5.0	18	8.0 cm	2	40.0 cm

5.3.3 Results

Fig. 5.5 shows a sequence of snapshots taken during three typical trials with radius factor $b = 1$, 2 and 4.

Influence of Size Ratio on Segregation Error

Fig. 5.6 shows a box-and-whisker plot [10] of the segregation errors for the different radius factors (b).

For $b = 1$, all e-pucks represented disks of identical size. Consequently, the segregation error (47.3%) was, on average, similar to the expected error for purely randomly distributed e-puck robots (50%). In no trial was perfect segregation observed.

For $b > 1$, the median segregation errors are all 0. The mean segregation errors are 1.31%, 0.07%, 0.49% and 0.28% for $b \in \{2, 3, 4, 5\}$, respectively. For $b = 2$, error free segregation was observed in 14 out of 20 trials. For $b \in \{3, 4, 5\}$, error free segregation was observed in 19, 18 and 19 of the 20 trials, respectively. That is, in these trials, all 20 e-pucks were spatially distributed as intended.

In 4 out of 60 trials for $b \in \{3, 4, 5\}$ the segregation was not error free. This was due to mechanical failures that caused robots to stop moving. For example, a robot became stuck on the arena floor, or lost contact with its battery.

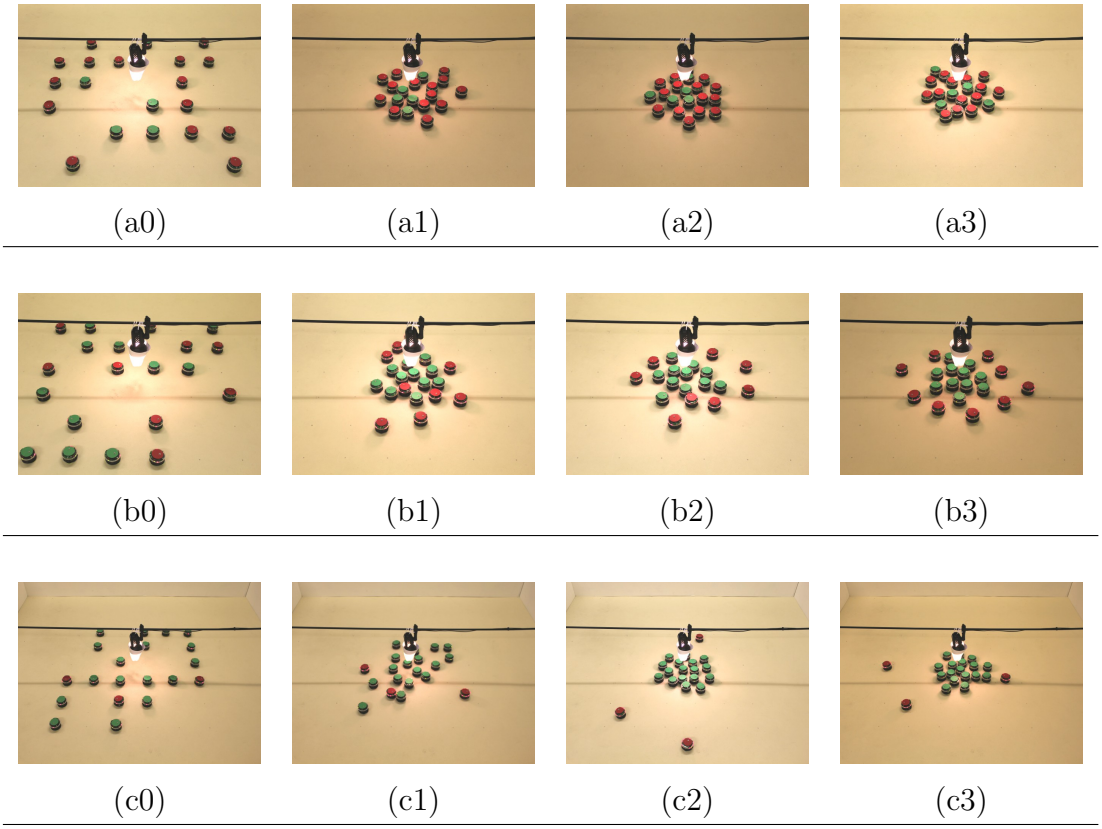


Figure 5.5: Sequences of snapshots taken during trials with radius factor b equal to 1 (top), 2 (center) and 4 (bottom). Robots with green markers represent disks of 8 cm radius. Robots with red markers represent disks of radius 8 cm (top), 16 cm (center) and 32 cm (bottom). The first and last images in each sequence (from left to right) show the initial and final configurations after 0 and 1200 sec. The other two images show intermediate situations.

Segregation Dynamics

Fig. 5.7 shows the segregation error over time as observed in trials with radius factor $b = 4$. Initially, the segregation error rapidly decreased until it became zero after 3.5 mins in most of the trials.

Influence of Size Ratio on Spatial Distribution

To understand better the effect of the size ratio (b), we analyzed the spatial distribution of robots of both groups. Fig. 5.8 shows the distances of all robots from the center of the arena as observed at the end of the trial. The data is grouped

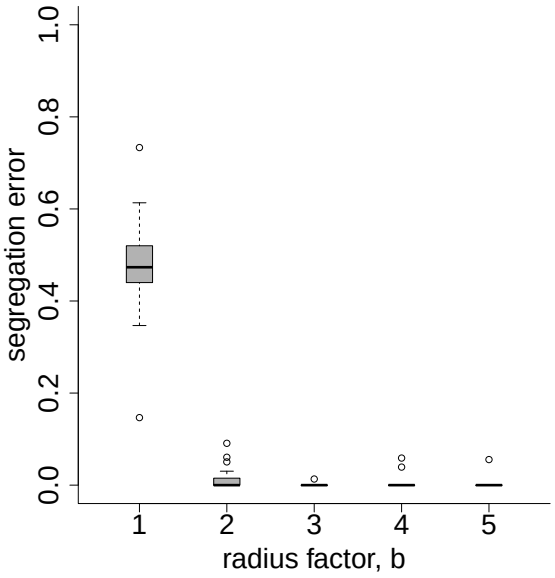


Figure 5.6: Box-and-whisker plot showing the segregation error observed in experimental trials with 20 e-puck robots for different radius factors (20 trials per radius factor).

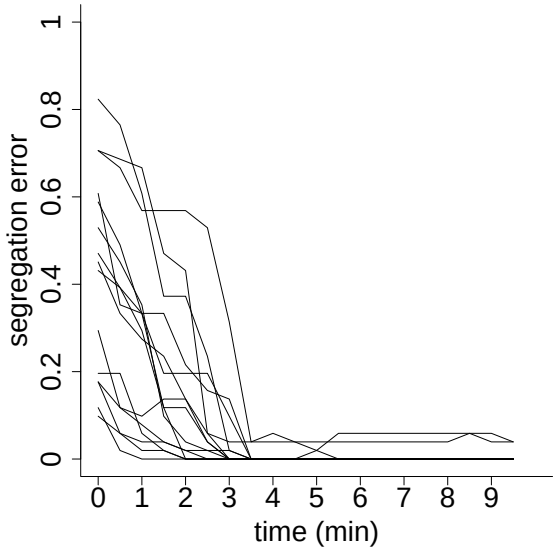


Figure 5.7: Segregation error over time for 15 experimental trials with 20 e-puck robots and radius factor $b = 4.0$. Data from the remaining five trials are not included because of some missing frames in the corresponding video recordings.

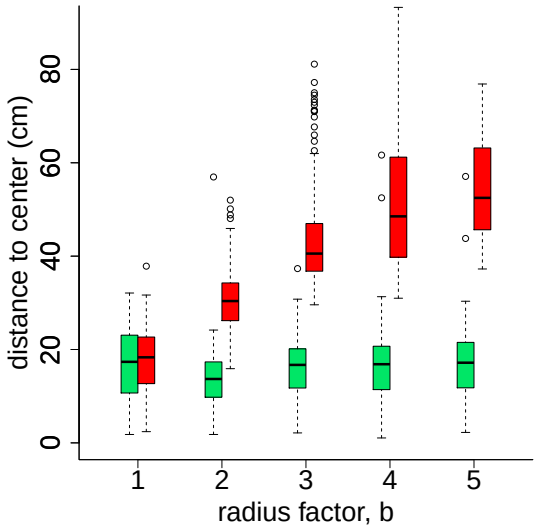


Figure 5.8: Box-and-whisker plot showing the distances of all robots from the center of the arena for groups of different radius factor (400 data points per radius factor). Green (light gray) boxes represent data from those robots that used the basic virtual radius, whereas red (dark gray) boxes represent data from those robots with the corresponding radius factor applied.

according to the two groups of robots presenting disks of different sizes in addition to the radius factor used.

For $b = 1$, robots of both groups were similarly distributed in space. The mean distances from the center of 'smaller' robots (green marker) and 'larger' robots (red markers) were 16.9 cm and 17.5 cm, respectively.

As b increased, the distance between robots of different groups increased.

For robots representing small disks (of 8 cm radius), the mean distance from the center of the arena mainly depends on the number of disks of that size. The largest number of small disks was present for $b = 1$ (in this case, all 20 robots were identical). For $b \in \{2, 3, 4, 5\}$, the numbers were 11, 15, 17, and 18, respectively (see Table 5.1).

The mean distance of 'larger' robots from the center grew almost linearly with the radius factor, setting them spatially apart from the other group. This caused the segregation error to decrease.

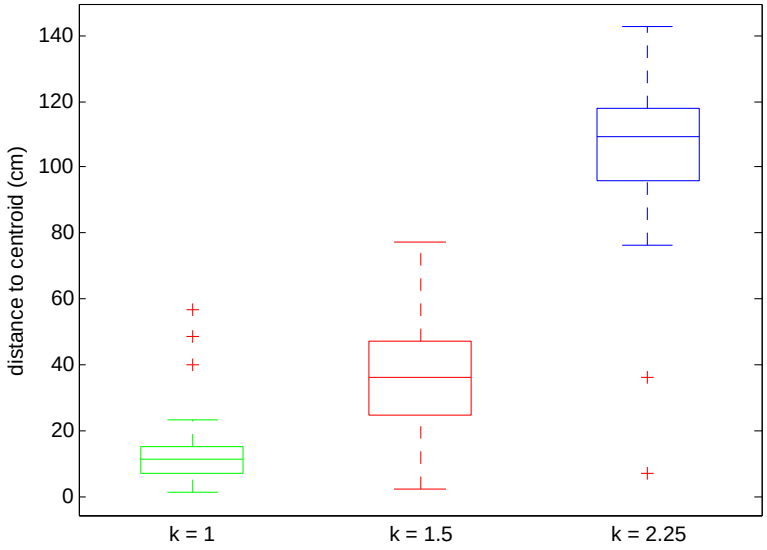


Figure 5.9: Box-and-whisker plot showing the distances of all robots from the center of the arena for groups of different radius factor.

5.4 Experiments using Three Groups of Robots

The previous systematic experiments shows that the segregation pattern can be formed when the virtual radii applied on the two groups of robots are different while a radius factor of 1 (no difference between groups) brought no segregation effect. The controller should also be effective when the number of groups in the formation is bigger than two, as long as the virtual radius of each group is sufficiently different.

Using three groups of robots ($m = 3$), 20 trials were performed in the same environment as the previous experiment. The number of robots in each group were 9, 12 and 9. The base virtual radius was 8 cm. The radius factor was $k = 1.5$. Thus, the virtual radii of the three groups were 8 cm, 12 cm and 18 cm (see Equ.(5.3)). The duration of a trial was 30 min.

The mean segregation error of the 20 trials was 0.288. Typical final patterns observed in these trials are shown in Fig. 5.10. The distance between robots of different groups to the centroid of the formation is shown in Fig. 5.9. These result shows that the controller is still effective when there are more than two groups of robots.

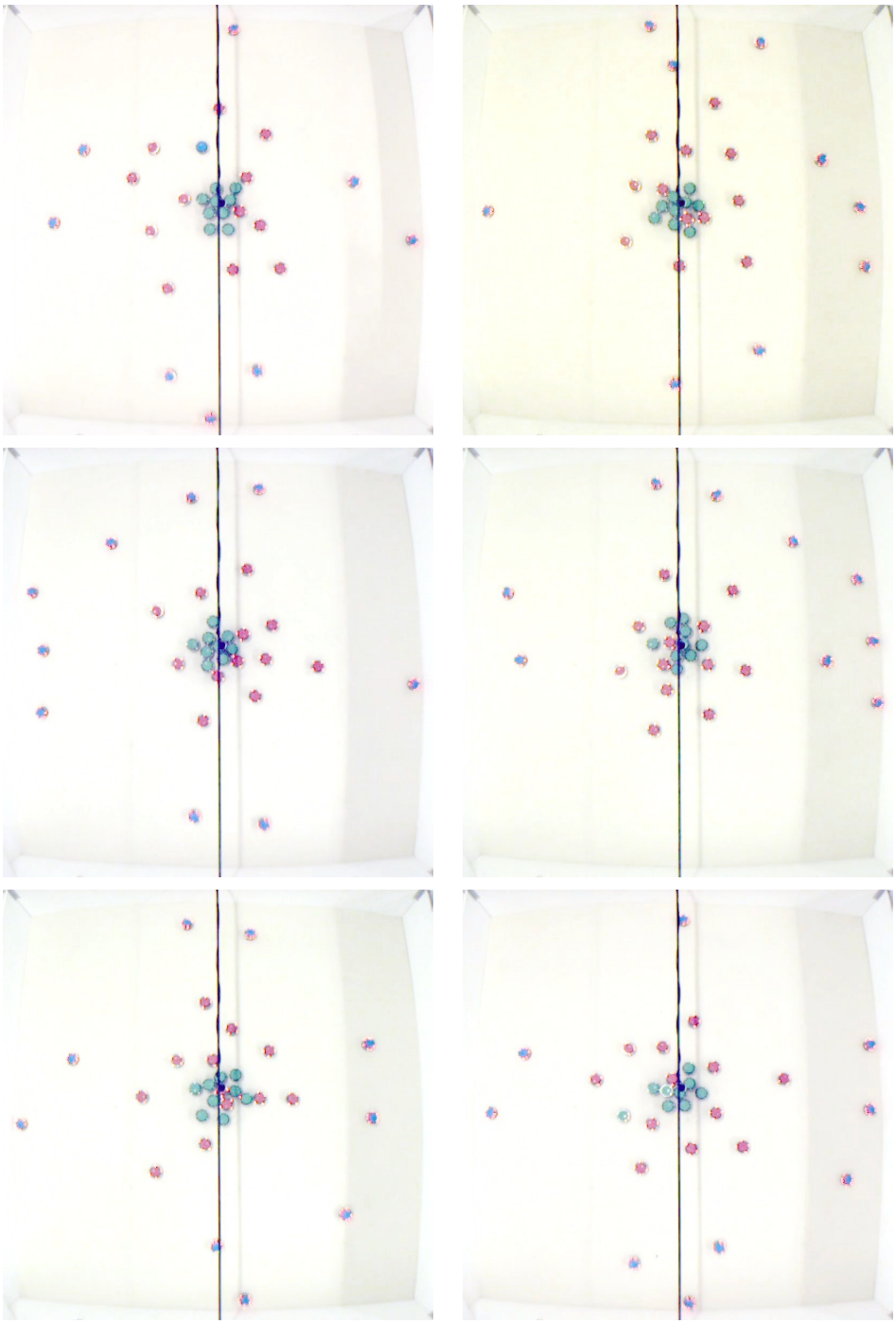


Figure 5.10: Positions of the robots at the end of 6 out of the 20 trials with three groups of robots. Robots with green, red and blue markers had a virtual radius of 8 cm, 12 cm and 18 cm, respectively.

Chapter 6

Fully Decentralized Segregation

In Chapter 5, a decentralized controller was implemented to organize a swarm of e-pucks that allows e-pucks to mimic the behavior of particles. We showed that a swarm of robots representing particles of 2 to 3 different size can segregation in to an annular pattern. The motion of a robot was governed by a combination of three components:(i) attraction towards a global location, which emulates the effect of a gravitational pull, (ii) random motion, which emulates the effect of vibration, and (iii) repulsion from nearby robots, which emulates the effect of collisions between the particles. Although this controller only used information from a robot's own sensors, the presence of a global reference point and the assumption of all robots to perceive it may not be met in practical applications.

In this chapter, we introduce a new algorithm to form an annular structured segregation that does not require a global reference point. In this algorithm the motion of a robot is still based on the three components in the Brazil nut effect. However, the attraction towards a global location was achieved through a collective signaling mechanism that does not involve any global information (see 6.1). Focusing on this new attraction mechanism, a simulation was implemented to study the performance of this controller.

6.1 Controller Design

The e-puck robot has limited capabilities but its architecture is very typical. Therefore, this segregation controller is design based on the model of e-puck. This means:

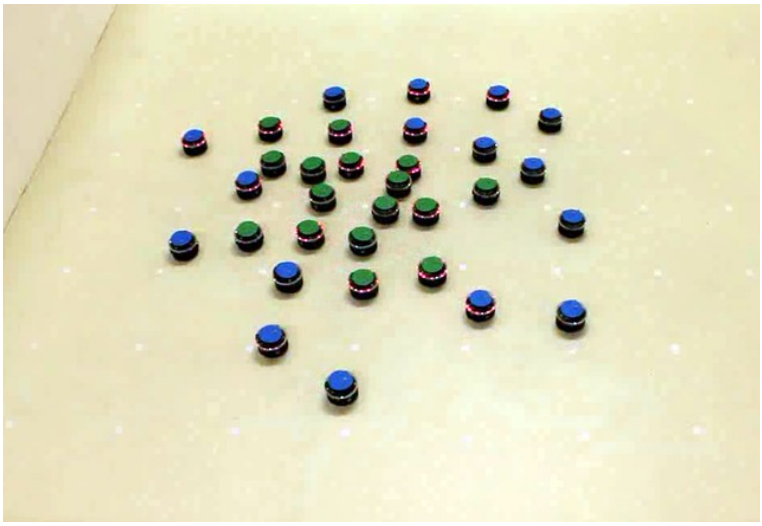


Figure 6.1: Segregation of 32 robots into an annular structure with two layers. There is no center of attraction (global reference point) in the environment. Instead, the robots are attracted by those robots that turn on their red LEDs. The red LEDs on all the robots are turned on and off according to a specific mechanism that was designed for achieving such a segregation. It is the main focus of this chapter.

the controller takes proximity sensors and directional camera as inputs to the while the motion of the robot only depends on the differential wheeled drive mechanism.

In the segregation controller introduced in Chapter 5, a robot needed to scan its surroundings through eight discrete rotation steps. In the controller presented here, such omni-direction perception is achieve through a continuous rotation, which means the robot keeps gather information from its directional camera while maintain a constant angular speed. The repulsive motion and attractive motion required in the Brazil nut effect are imposed on the rotational motion. However, these two effects do not affect the angular speed severely so the robot can still perceive information from all directions.

In the controller presented in Chapter 5, the random component and the repulsion component in the Brazil nut effect had already been implemented in a decentralized manner. Therefore, the challenge to design a fully decentralized segregation controller lies in the attraction component.

Two types of attraction mechanisms are designed. They are referred to as RANDOM PROPAGATION (RP) mode and EQUAL ATTRACTION (EA) mode. RP mode is a relatively novel mechanism that takes inspiration from those reaction-

diffusion systems in nature that are able to form apparent patterns [130] [76]. For example, the zebra stripes or the giraffe pattern are both patterns formed through local interactions of skin cells. EA mode employes a basic method that originates from aggregation behaviors of a swarm of mobile agents. Its purpose is to serve as a benchmark for comparison with the RP mode.

The RP mode is the focus of this chapters. When the decentralized segregation behavior is referred in a general context (e.g. in other chapters), it means the controller based on RP mode.

EQUAL ATTRACTION mode

The apparent consequence of the attraction effect in the segregation behavior in Chapter 5 is to aggregate the robots. Such aggregation effect was achieved by making all of the robots taxis towards a global infrared light source. Therefore, it is possible that any aggregation behavior can be applied to substitute the role of the global center of attraction in the Brazil nut effect. Following this idea, EQUAL ATTRACTION mode was designed.

Aggregation is a well studied behavior in swarm robotics; many solutions and phenomenons have been published [92] [72] [128]. One basic decentralized method to achieve aggregation is to make each robot move towards its neighbor. As a consequence of such a behavior, the swarm of initially separated robots may come together.

Such a simple aggregation method has its flaws in an aggregation task. For example, if the initial distribution of the robots is non-uniform, this behavior may lead to a partial aggregation in which the separated robots form a number of clusters instead of one clusters [42]. Because this project is not focusing on the aggregation behavior, this method can serve its purpose to keep the group of robots aggregated.

RANDOM PROPAGATION mode

From the perspective of a segregation controller, the EA mode can be viewed as each of the robots having an equivalent attraction to its nearby robots.

In RANDOM PROPAGATION mode, the robots are also attracted by each other rather than a global reference point. However, the attraction of a robot

towards the robots around it does not constantly exist. Instead, the robot is attracted only by those robots that have a specific signal activated. The effective range of such signals are not global. The signal can be observed by the other robots within a limited range and it can be occluded by the robots.

These signals appear dynamically among the robots. When a robot does not have the signal activated, two events will make it activate the signal:

1. The robot has a certain probability to activate the signal. This is the rule of random activation.

2. When it perceived a signal from the other robots. This is the rule of propagated activation.

Once the signal is activated, it will be deactivated automatically after a period. This period is called the signal period. Within this period, the robot will contribute an attraction effect towards the robots that can perceive the signal. Furthermore, once the signal is activated, the robot will ignore any of the activation rules stated above for a period that is longer than the signal period. This period is called the inhibitory period.

In the description, the rules of the signal activation will be referred to as the signal propagation mechanism. The event of activating a signal will be specifically referred as *fire* while the signal it-self with just be referred as the *signal*. Thus, if a robot *fired*, the other robot may be able to perceive the *signal* of the robot.

6.2 Controller Implementation

The segregation controller was implemented using the simulation framework introduced in Section 3.2 and ported on to physical e-pucks. Because the systematic experiments in this study had been done in computer simulations, this section will introduce the details of the implementation based on the version used in the simulation. However, the description in this section is also valid for the version on the physical e-pucks that are used to do demonstration trials showing in Fig. 6.1.

6.2.1 Motion Controller

The motion of the robot is controlled by a reactive controller, in which the wheel speeds of the robot is calculated in each time step according to the sensor inputs.

The basic control framework is similar to the one used in the motion control of the cooperative transport controller introduced in Section 4.4.2. The left and right wheel speeds (ω_l and ω_r) are calculated from the linear speed, S, and the angular speed, D, using the following equation:

$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1 & -s \\ 1 & s \end{bmatrix} \begin{bmatrix} S \\ D \end{bmatrix}, \tag{6.1}$$

$$s = \begin{cases} 1 & \text{if } S \geq 0, \\ -1 & \text{if } S < 0. \end{cases} \tag{6.2}$$

In this controller, the linear speed (S) is calculated by

$$S = S_{attract} + S_{repel} + S_{random} + n_S. \tag{6.3}$$

In the above equation, $S_{attract}$, S_{repel} and S_{random} are the linear speed augmentations to impose the attraction, repulsion and random motion in the Brazil nut effect respectively. n_S is the neutral linear speed.

The angular speed, D, is calculated by

$$D = D_{repel} + n_D. \tag{6.4}$$

In the above equation, D_{repel} is the angular speed augmentation to impose the repulsion. n_D is the neutral linear speed.

The two neutral speeds are constant values. They determine the motion of the robot when it is not affected by anything in the environment. The design of the controller makes the robot rotate on its position so it can scan the surrounding using its directional camera. This is realized by setting n_S and n_D to 0 and 0.4 respectively.

The augmentation values represent the effects of the three motion components required by the segregation behavior. Among them, the random component is simply achieved by adding S_{random} to the linear speed. S_{random} is a value that is randomized once every T_r second in the range of $[-M, M]$. The attraction components and repulsion components are determined according to sensors inputs,

they will be introduce in detail in next sections. All constant parameters involved in the expressions can be found in Table 6.2.

Attraction Augmentation

If a robot perceives signals from another robot, the robot is attracted by that robot. The signal is implemented using the red LEDs on the e-puck. The robot uses its camera to perceive any firing robots in front of it. The presence of firing robots can be judged from the sensor output w_r provided by the camera image processing routine introduced in Section 3.1.2.

When the robot perceived the signal from another robot, it will set $S_{attract}$ to a positive value. Because the robot is more or less facing the firing robot, the forward motion brought by this change is essentially attracting the robot to the firing robot.

The pseudocode of attraction mechanism is given in Algorithm 4.

Algorithm 4 The Attraction Augmentation in The Motion Controller

```
IF w_r > H_s THEN
    SeesSignal = TRUE;
ELSE
    SeesSignal = FALSE;
ENDIF
IF SeesSignal is TRUE THEN
    S_attract = C_a;
ELSE
    S_attract = 0;
ENDIF
```

Repulsion Augmentation

In this segregation controller, the proximity sensors on the robots are working in the EMITTER ON mode. In this mode, the presence and the proximity of nearby robots can be measured in the ambient samples of the proximity sensors, **A** (see Section 3.1.3).

Table 6.1: Weight of Proximity Sensor Inputs

Sensor	#0	#1	#2	#3	#4	#5	#6	#7
$\mathbf{D_A}$	2.25	1.75	1.5	1.5	-1.5	-1.5	-1.75	-2.25
$\mathbf{S_A}$	-0.25	-0.25	0.125	0.5	0.5	0.125	-0.25	-0.25

The repulsion terms, S_{repel} and D_{repel} are calculated based on $\mathbf{A}$:

$$S_{repel} = k\mathbf{S_A}\mathbf{A}. \tag{6.5}$$

$$D_{repel} = k\mathbf{D_A}\mathbf{A}. \tag{6.6}$$

The value k is the factor of the repulsion augmentation. Different groups in the segregation behavior have different repulsion factors. After the behavior reaches stable stage, the group with a higher k is supposed to be further outside in the formation than those with lower k's.

The weighting matrices $\mathbf{S_A}$ and $\mathbf{D_A}$ are given in Table 6.1. This configuration makes the robot move away nearby robots or obstacles. For the linear speed, $\mathbf{S_A}$ matrix contributes a positive term when there are things on the robot's back and negative term for things in front of the robot. Therefore, this controller let the robot 'bounce' away from other robots in front or behind. For the angular speed, this matrix generates a angular speed augmentation that turns the robot away from nearby objects to achieve the repulsion effect.

6.2.2 Signal Propagation Mechanism

To implement the EA mode, the robot keeps the red LEDs turned on constantly. Because the motion controller make the robot move towards any red LEDs, the robot is attracted by all robots it sees in this mode.

In the RP mode, a robot that has not fired for a certain period will fire randomly by chance or when it sees a firing robot.

These rules are implemented accordingly. The pseudocode of the implementation is given in Algorithm 5. Variable `InhibitoryTimer` is the timer that memorize the time after the last firing. Variable `SignalTimer` is the timer that is used to keep the red LEDs turned on for a period of time.

Table 6.2: Parameters in the controller

Symbol	Value	Description
T	1/60	step size of a time step in second
T_s	6	signal period in number of time steps
T_h	7	inhibitory period in number of time steps
T_r	60	the interval in time steps that S_{random} is randomized
M	0.064	random span of S_{random}
C_a	0.4	the speed augmentation when the robots see a firing robot
p_{init}	0.01	probability to initiate a signal firing in each time step
H_s	3	the pixel count threshold to judge the presence of any signals
k	$\in [1, 8]$	repulsion factor of a group

The code to set the attraction motion augmentation ($S_{attract}$) is also included in Algorithm 5 to clarify its relation to the signal propagation mechanism. Therefore, Algorithm 5 can be viewed as the entire distributed attraction mechanism purposed in this fully decentralized controller.

6.3 Computational Experiments

In order to further study and compare the two modes of the controller, systematic experiments were conducted. Unlike the cooperative transport system, this system does not involve many physical interactions. Ideally, no collisions between robots should happen. Thus, the accuracy of the collision physics in computer simulation would not significantly affect the experimental result. Therefore, the simulated system was also used to conduct systematic computational experiments.

6.3.1 Setup

Three groups of robots were used in the experiments. The only difference between the robots in different groups was the repulsion factor, k, applied in their controllers. The number of robots in each of the groups and their common repulsion factor are listed in Table 6.3.

In preliminary tests, it was shown that the final formation of the swarm tends

Algorithm 5 The Attraction Mechanism

```
IF w_r > H_s THEN
    SeesSignal = TRUE;
ELSE
    SeesSignal = FALSE;
ENDIF

IF SeesSignal is TRUE THEN
    S_attract = C_a;
ELSE
    S_attract = 0;
ENDIF

IF InhibitoryTimer > 0 THEN
    InhibitoryTimer = InhibitoryTimer - 1;
ELSE
    IF SeesSignal is TRUE OR Rand(0,1) < p_init THEN
        InhibitoryTimer = T_h;
        SignalTimer = T_s;
    ENDIF
ENDIF

IF SignalTimer > 0 THEN
    Turn On All LEDs;
    SignalTimer = SignalTimer - 1;
ELSE
    Turn Off All LEDs;
ENDIF
```

Table 6.3: The Specifications for Each Group

Group	k Value	Group Size	Marker Color
Group A	2.0	19	Orange
Group B	4.0	18	Blue
Group C	8.0	24	Green

to be circular (e.g. Fig. 6.2(b)). Therefore, the number of robots in each group was chosen according to the ideal number of robots to form certain layers in a hexagonal packing arrangement of circles [45]. For instance, Group A is supposed to be the first three layers in such an arrangement, which ideally needs $(1+6+12)$ robots.

The robot model used in the simulation was again the e-puck, which has a diameter of 7.0 cm. The environment was a square arena of size 400 cm $\times$ 400 cm. The initial positions of the swarm of 61 robots were randomly chosen from an 8×8 grid that has a total dimension of 150 cm $\times$ 150 cm and is centered in the arena. The orientation of the robots are uniformly random. A typical result of such a random initial formation is shown in Fig. 6.2(a).

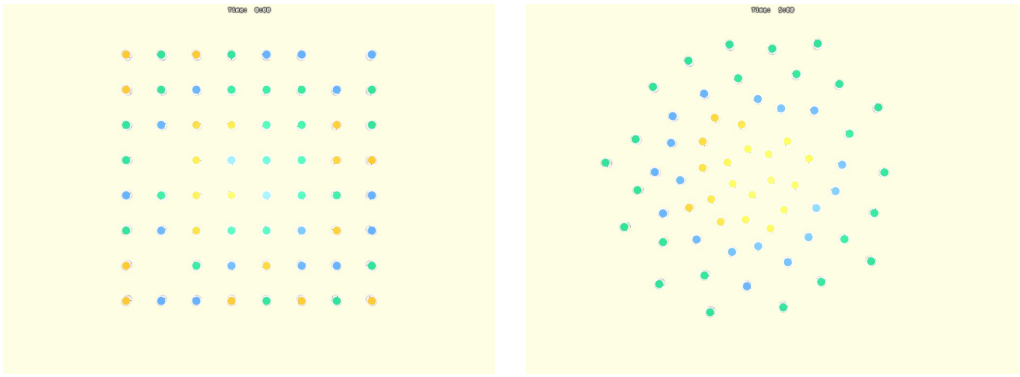
In preliminary tests using the setup introduced above, a visible segregation pattern may occur for both of the modes after 150 sec. Thus, the duration of a trial in the systematic experiment was set to 300 sec. The simulation framework was running at 60 times steps per simulation second. The status of each robot was logged once every two time steps, so each robot generated 9000 data steps. For each of the RANDOM PROPAGATION mode and the EQUAL ATTRACTION mode, 30 trials were executed.

6.3.2 Results

Segregation Error

The metric used to the evaluate the degree of spatial segregation of the robots is the segregation error introduced in Section 5.3.2. This definition of the segregation error was first proposed in [51].

In this metric, the value 0.5 means the robots in different groups are well mixed, which implies no spatial segregation appears in the swarm. The value 0.0 means



(a) (b)

Figure 6.2: (a) A typical initial formation of the swarm in the systematic experiment. (b) A typical final formation of the swarm in RANDOM PROPAGATION mode.

the robots formed an annular structure in which the robots in a group with a higher repulsion gain are always further outside the groups with lower repulsion gains. This is the segregation error that this segregation behavior was supposed to achieve. The value 1.0 means that the robots formed a annular structure in which the robots in a group with a higher repulsion gain are always surrounded by groups of robots with lower repulsion gains. This is the opposite of what the segregation behavior was supposed to achieve.

The dynamics of the mean segregation error of the trials is shown in Fig. 6.3. Fig. 6.5 plots the segregation error over time for each of the trials. Fig. 6.3 also shows a second order exponential fit of each of the curves. For the EA mode, the fitted function is:

$$f(x) = 0.5391e^{-0.01707x} + 0.0009604e^{0.007137x}. \tag{6.7}$$

For the EA mode, the fitted function is:

$$f(x) = 0.1429e^{-0.04481x} + 0.3575e^{-0.00262x}. \tag{6.8}$$

Considering Fig. 6.5, the spatial segregation occurred in all of the trials in RP mode. Whereas, the trials for EA mode have a much slower progression of the spatial segregation.

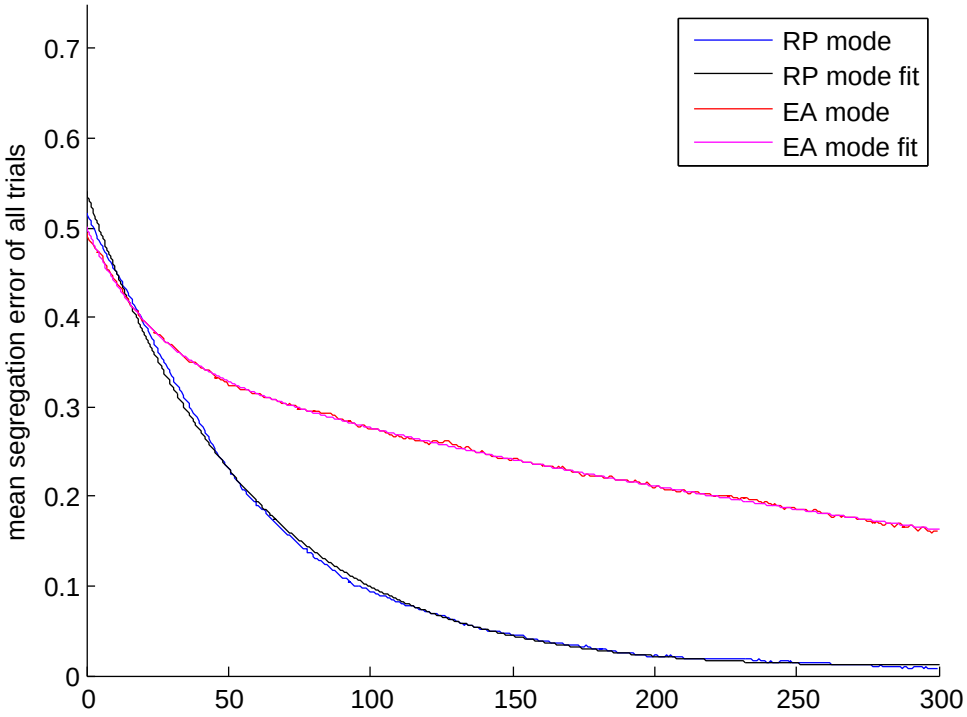


Figure 6.3: Segregation error mean plot of all of the trials in each of the modes, each fitted with a second order exponential function.

Group Distance

Let a robot's centroid distance be the distance between its position and the centroid of the formation. The group distance is then defined as the mean of the centroid distances of the robots in the corresponding group in a trial. Fig. 6.6, the group distances of the trials are plotted. The group distances are normalized so that the group distance of Group C is always 1.0.

From Fig. 6.6, it is clear that the group distances observed in the RANDOM PROPAGATION mode trials are much more stable than the group distances of the EQUAL ATTRACTION mode trials.

Firing Rate

Among all trials, a common phenomenon was observed in the firing pattern of the robots running the RP mode controller: the robots near the center of the formation fires more frequently. In order to verify this observation, the firing rate metric was

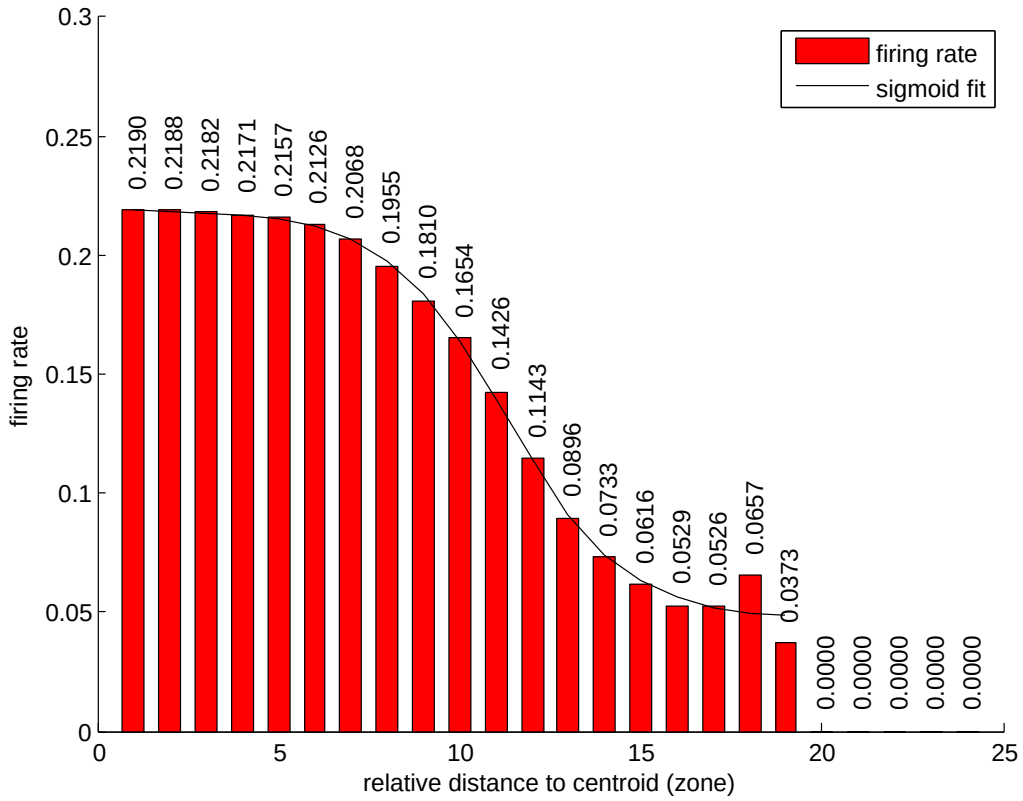


Figure 6.4: Firing rate of the zones for all trials of the RANDOM PROPAGATION mode, fitted with a sigmoid function. For details, see text.

designed.

To study the spatial distribution of the firing rate of the swarm, this metric is measured for different regions in the formation. Because the segregation behavior forms an annular structured formation, different regions are divided according to the robots' centroid distance. More particularly, a region in which the positions inside has a similar centroid distance is defined as a zone. As a result, the division of the zones in the formation looks like a marksman target. The function to judge which zone robot i is in is defined as

$$J(i) = ceiling(d_i/w), \tag{6.9}$$

in which d_i is robot i's centroid distance. w is the width of all zones, which is 7 cm.

The firing rate of zone j is defined as

$$q_j = \frac{E_j}{O_j}, \tag{6.10}$$

in which: O_j is the occurrence count of the robots in zone j. In every time step, O_j will be increased by 1 for each of the robots occurred in zone j. E_j is the firing count of the robots in zone j. For each firing event occurred in zone j, E_j will be increased by 1.

Fig. 6.4 shows the bar plot of the firing rate distributions. In this plot, E_j and O_j was counted across the full durations of all trials of the RP mode[1].

It can be observed that the firing rate of the robots are almost monotonically increasing when the distance to the centroid is decreasing, which is far different from the firing rate in the random initiation rules defined in the signal propagation mechanism. This could probably be one of the explanations of why RP mode performed much better than EA mode in the systematic experiments. In Fig. 6.4, the distribution of the firing rate against zone is fitted with the following sigmoid function:

$$f(x) = 0.2190 - \frac{0.1721}{1 + 0.2624 * 10^{(11.2471 - x)}} \tag{6.11}$$

[1]due to how the EA mode is implemented, 'firing rate' of the EA mode trial would be constantly 1 for all zones.

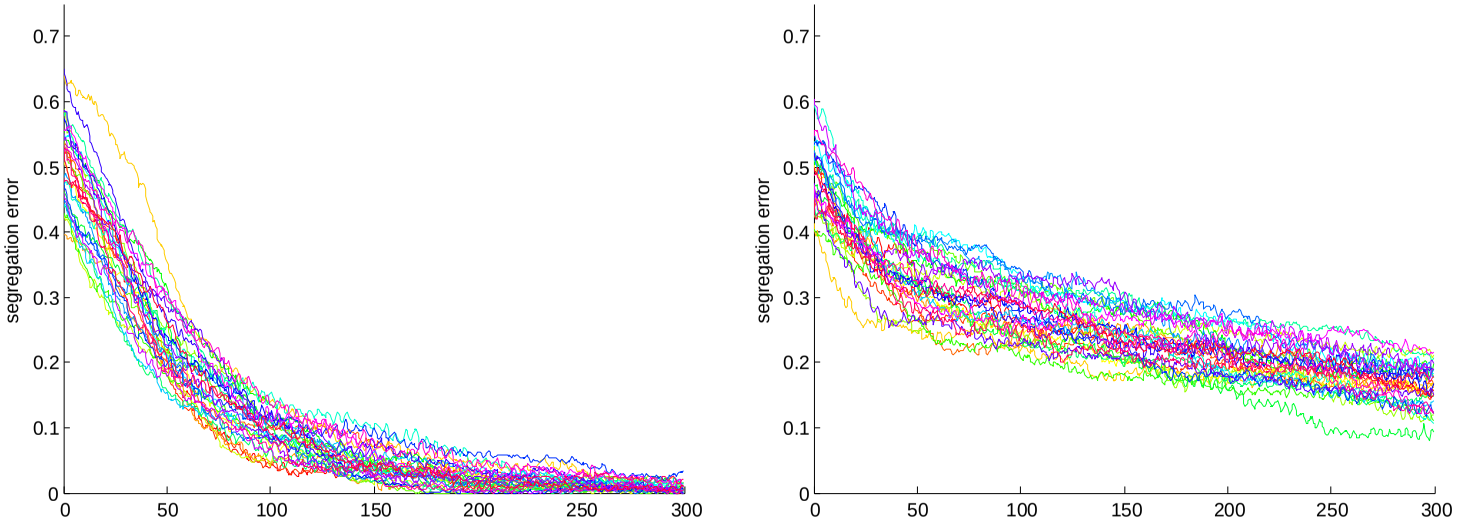


Figure 6.5: Segregation error plot of the 30 trials. (a) For RANDOM PROPAGATION mode. (b) For EQUAL ATTRACTION mode.

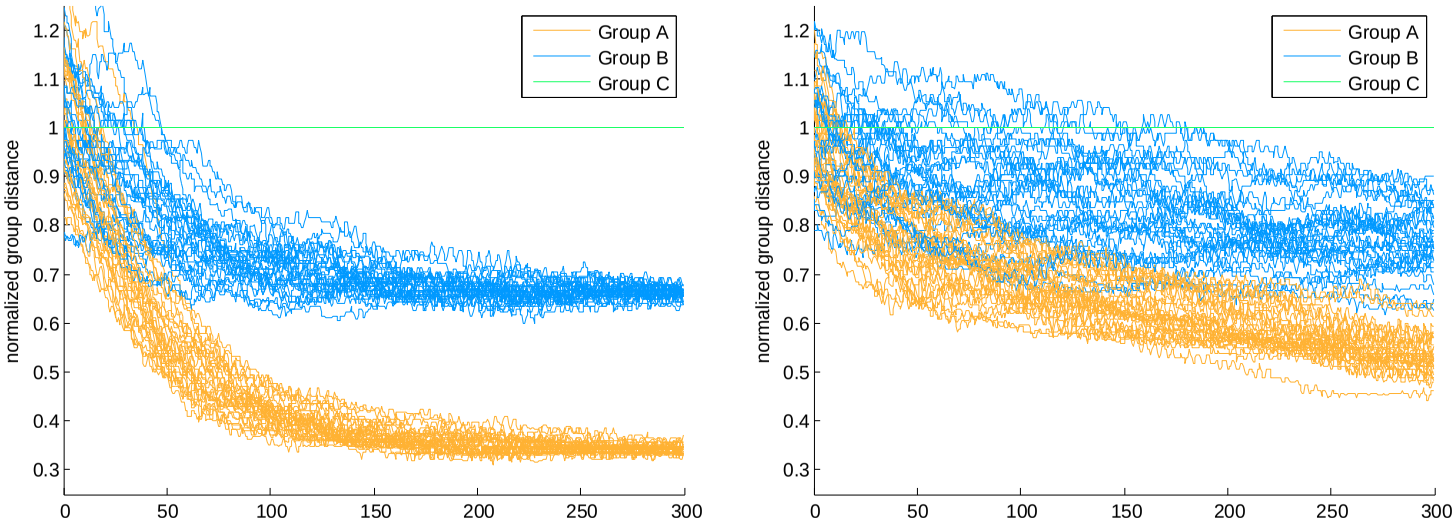


Figure 6.6: Normalized group distance plot. The group distance the mean of the distance to the formation centroid of those robots in the group. Each one of the curves in the plot is the time series of the group distance in a trial. The distances is normalized according to group C's distance. (a) For RANDOM PROPAGATION mode; (b) For EQUAL ATTRACTION mode.

6.3.3 Discussion

The performance of the two modes in the systematic experiments were evaluated using both of the segregation error and the group distance as the performance metrics. According to the metrics, spatial segregation successfully appeared. However, the RP mode controller performed much better than the EA mode controller.

Videos recording the typical situations in the computational experiments can be found in [25]. The RP mode controller has also been ported to physical e-puck robots. A demonstration trial using 32 e-pucks divided in to two groups has been performed successfully. Video of this trial are also included in [25].

The only difference between the two modes was the method to simulate the gravitational attraction in the original Brazil nut effect. The significant difference in the performance of the two modes shows that the signal propagation mechanism is much more effective than the uniform attraction mechanism employed from a common aggregation behavior. However, the reason behind this was not very obvious. Two hypothesis had been raised:

1. The attraction strength caused by the two mode is different. Generally, the attraction augmentation in EA mode is constantly existing (non-zero) while it exists in RP mode sometimes only. A lower attraction strength of RP mode may improve the segregation performance over the EA mode.

2. The distribution of the firing rate caused by the emergent behavior of the signal propagation mechanism generates a attraction towards the center of the formation for all of the robots. Comparing to the uniform adhesion between the robots in EA mode, this centripetal motion trends is more similar to the attraction effect in the original Brazil nut effect. Therefore, the signal propagation mechanism can serve its role in the segregation behavior better.

The later case has been studied further because of the distribution of the firing rate unveiled in Fig. 6.4. In order to fully understand the cause-and-effect of this phenomenon, some further investigations focused on it have been done. The next section will introduce these investigations.

6.4 Further Investigation

6.4.1 Visualizing the Gradient of Firing Rate

In order to observe the firing pattern of the robots better, an extra trial was done using the same configuration used in Section 6.3. During this trial, screenshots were captured at different stages of the trial. These screenshots are shown in Fig. 6.7. In this figure, it can be observed that: regardless the segregation stage, the center area is always illuminated better than the area near the edge.

Fig. 6.8 is the result of super-positioning all of images in Fig. 6.7. This is achieved by adding the red channels of the images and then normalizing the sum. Essentially, it is the same as normalizing the mean of the red channels of the images. According to Fig. 6.8, it is clear that the center area has a higher mean. Combined with Fig. 6.4 and Fig. 6.7, the firing rate of the robots has a radial gradient in the circular formation.

In the implementation of the fully decentralized controller, the signal propagation mechanism barely has any cohesion with the motion of the robots. For example, it only requires the motion of robot to have an angular speed so that the robot can use its directional camera to scan the surrounding. Therefore, it is very possible that the signal propagation mechanism alone caused the radial gradient.

6.4.2 A Cellular Automaton Based Simulation

A radial gradient appeared in the firing rate of the fully decentralized segregation behavior. In order to study whether this phenomenon was related to only the signal propagation mechanism or the whole segregation controller, the signal propagation mechanism was abstracted to a set of rules that can be applied as a cellular automaton. This means there are no motion or sensing problems involved.

The firing rule applied on individual cells is very similar to the signal propagation mechanism in the segregation controller. An individual cell has a certain probability of firing. This is the rule of random activation. In each time step, the cell will also randomly selected one of its four adjacent cells. If this selected adjacent cell fired, the cell will fire too. This is the rule of propagated activation. Note, the cell will only read the randomly selected neighbor cell in each time step, which is originated from the directional perception of the signals in the segregation

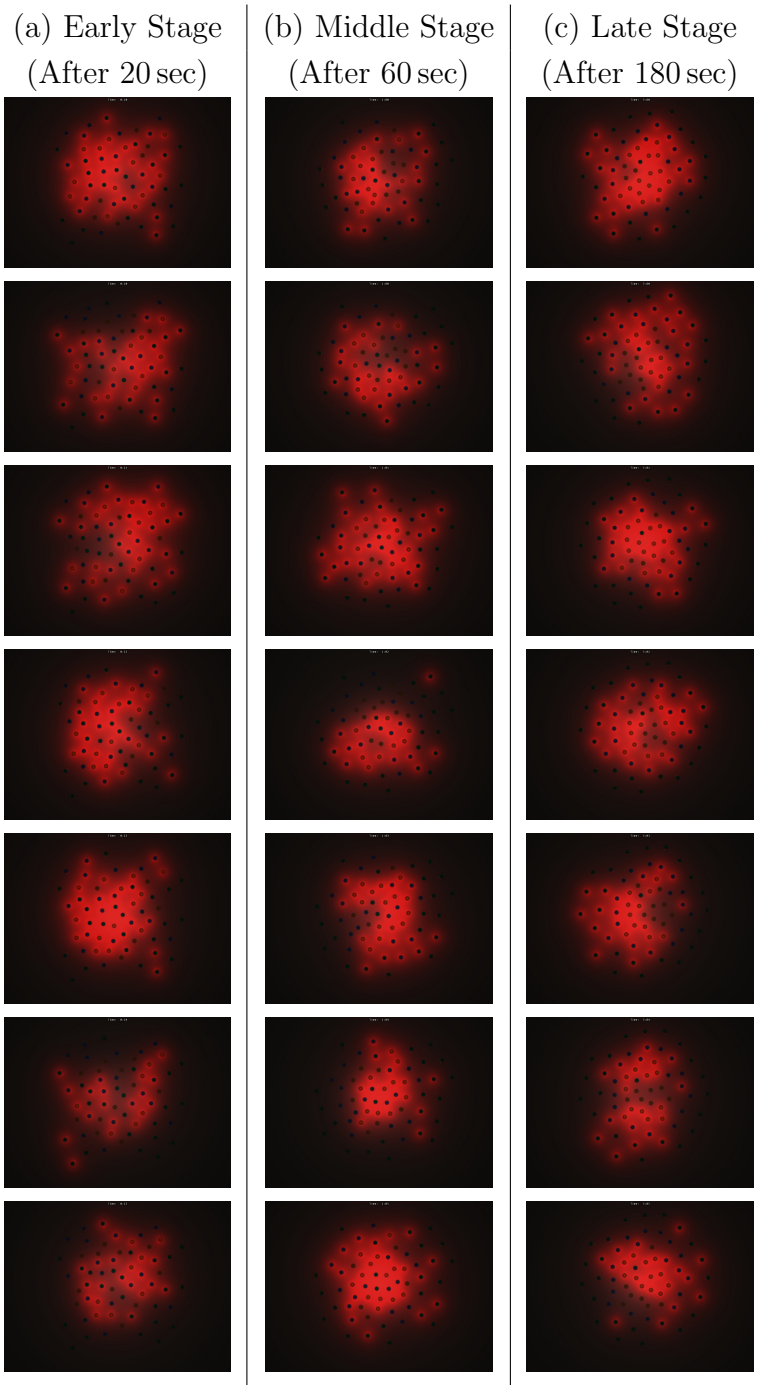


Figure 6.7: Snapshots showing three stages in a simulation trial. The images were captured every 53 time steps after the first image (top row) in the corresponding stage. The prime number 53 was used as the capture interval to prevent alias. It can be observed that regardless the segregation stage, the center area is typically better illuminated by the LEDs than the area near the edge.

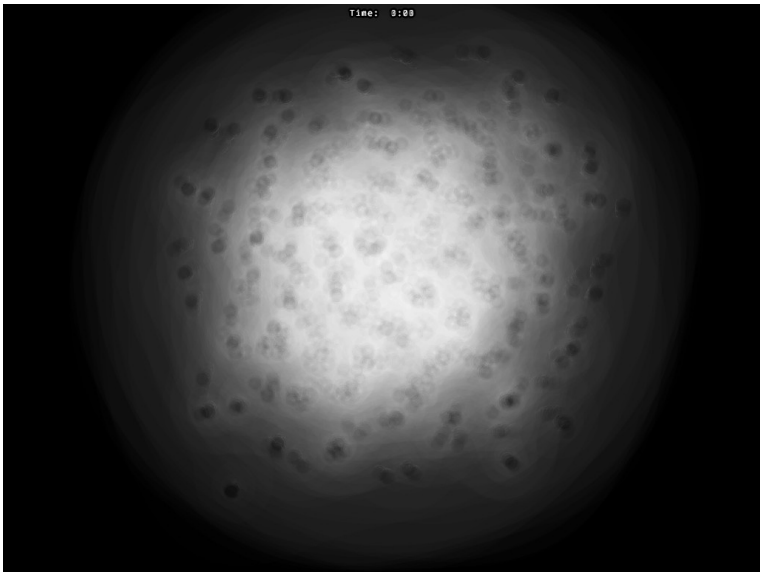
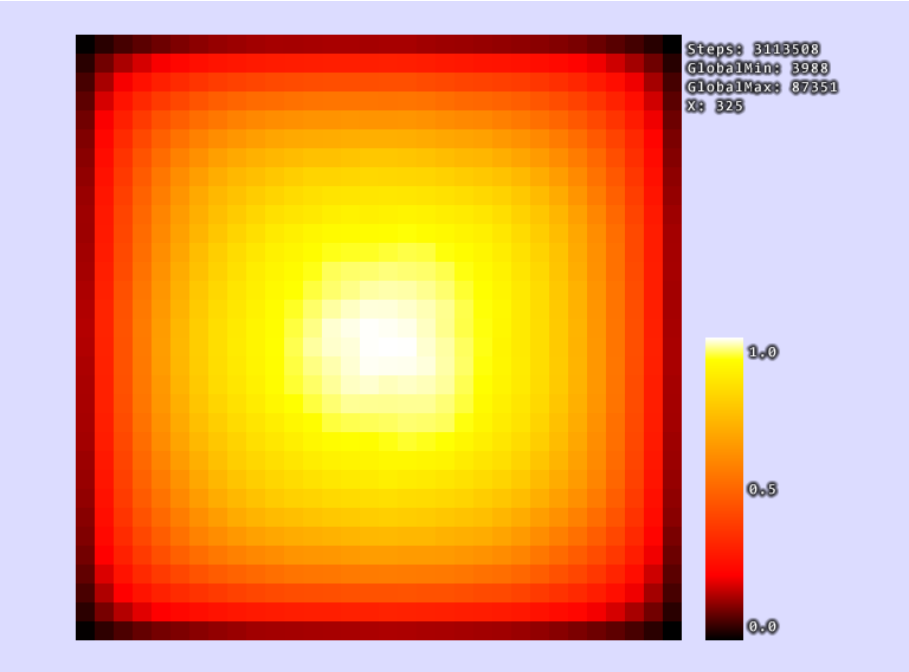


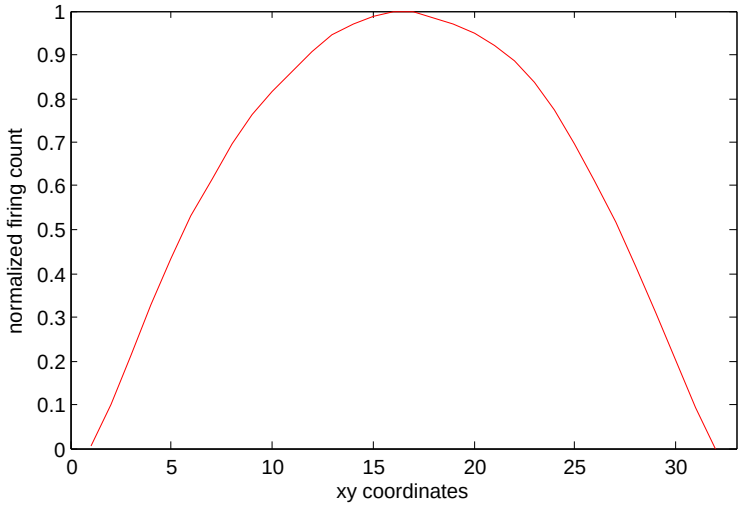
Figure 6.8: This gray-scale image was made by normalizing the addition of the red channels of the 21 images in Figure 6.7. It can be observed that the lighting caused by signal firing if the robots had a radial gradient. This phenomenon might be caused by two situations: (1) the spatial distribution of the robots near the center had a higher density. (2) the robots near the center were firing more frequently. According to the observation of the final formation, the first situation was not convincing. Therefore, the second situation was investigated further.

controller. This is unlike typical cellular systems in which each of the cells usually take inputs from all of its adjacent cells (e.g. 8 for Moore neighborhood and 4 for von Neumann neighborhood). After a firing event, the fired cell will enter a inhibition period in which it will not use either of the firing rules.

The plot of the firing counts of a trial is given in Fig. 6.9. The whole firing count landscape of the cellular automaton (Fig. 6.9(a)), reveals a centered radial gradient appear among the cells. In this cellular automaton, the cells did not move and they only read from their adjacent cells. Yet, an apparent global pattern formed. This result could support the hypothesis that the firing rate gradient caused by the signal propagation mechanism is independent from the segregation controller.



(a)



(b)

Figure 6.9: The normalized firing counts of the cells. This firing counts are normalized so the cell fired most has a score of 1.0 whereas the cell fired least has a score of 0.0. (a) the entire landscape of the 32×32 automaton; (b) the scores of the 32 cells along the $y = x$ diagonal on the plane.

Chapter 7

Conclusion

Inspired by social insects in nature, swarm robotics aims to use a large number of relatively simple robots to solve complex tasks through cooperation. In a swarm robotic system, the individual robots are typically simple. Their behaviors often rely only on local information which is collected directly from their own sensors. Such simplicity may bring the system advantages in robustness, flexibility and scalability. However, under these assumptions, it is challenging to coordinate large numbers of individuals to achieve tasks that are meaningful from a global perspective. For example, organizing a group of robots into a formation is difficult to achieve when there is not a common global reference frame.

Focusing on groups of miniature mobile robots without explicit communication capability, this thesis studied the problems of cooperative transport and pattern formation using fully decentralized controllers. This chapter summarizes the outcomes as well as provides directions for future work.

7.1 Summary of Outcomes

The thesis considered three controllers for two tasks. In the following, the outcome and directions for future work for these works are discussed.

Cooperative Transport Based on Occlusion

This study introduced a cooperative transport strategy that uses a large number of relatively simple and small mobile robots to transport a large object that can

occlude the robots perception of the goal. The strategy makes robots push along the surface of the object where the robots' line of sight to the goal is occluded by the object itself. By ensuring that the robots only push the object over the occluded surface, the object will eventually reach the goal (but the orientation of the object can not be controlled). A mathematical formulation of the strategy in a 2-D work space was provided. We proved that any convex-shaped object will always be successfully transported to the goal point and that the same is not necessarily true for objects of concave shape.

The strategy was implemented on the e-puck platform. Systematic experiments were performed to verify the implementation using three particularly challenging types of objects. In 43 out of 45 trials in total, the objects were successfully transported to the goal by 20 e-pucks. The self-correction effect introduced by the occlusion-based strategy can be clearly observed in these trials. Depending on the shape of the objects, the paths traced were on average 9.5% to 32.6% longer than the shortest possible path. The paths were also compared against predictions from the mathematical model. While most individual paths differed substantially, their overall distribution showed a good correspondence in both visual inspection and the path efficiency comparison. In an extended experiment, an extra e-puck was used as the goal. This goal robot was remotely controlled by a human operator. Following the path of the goal robot, the transport robots pushed the object in all 20 trials through an environment with obstacles.

A physics-based simulation was used to show an implementation of the transport strategy in a 3-D environment using a swarm of conceptual robots that have only four binary sensors. The simulation shows that the transport strategy has potential to be implemented in a 3-D environment using a large swarm of simple robots. For example, nano-robot swarms could transport materials such as drugs within the human body.

To the best of our knowledge, this is the first successful attempt of using a large number of autonomous robots to push a large non-specific object. Moreover, the experiment using a mobile goal can be viewed as a successful instance of human-robot interaction in which a human remotely controls a swarm of robots through a single agent robot. Finally, we have shown the potential for the strategy to be applied in real world environment, through a proof-of-concept experiment in a domestic environment.

Segregation Based on the Brazil Nut Effect

An algorithm inspired by the Brazil nut effect from computer simulations [51] was ported to the e-puck robot. The algorithm lets a group of robots mimic a mixture of embodied particles in 2-D space (disks) under vibration and centrifugal gravity. Each of the robots assumes the other robots have the same disk radius as itself, so the robots do not need to communicate the disk radius with each other.

The original algorithm in [51] assumed that every robot can instantly measure the relative position of all the robots in its vicinity. Here, we showed how this algorithm can be modified to allow for an implementation using directional vision. This implies that (i) robots have to revolve in order to obtain an omni-directional picture and (ii) the algorithm has to cope with misperceptions, for example, due to visual occlusion. The robot also needs to sense the relative direction to a point of attraction in the environment (to emulate the effect of a gravitational pull). Here, we used a light bulb, which was perceived by the e-puck's infrared sensors.

We believe that the new algorithm is applicable to a wider range of robotic platforms when compared to the original algorithm. In principle, the new algorithm can be implemented on any wheeled robot with a camera or equivalent sensor to detect nearby robots. Note that in principle, the light bulb could also be perceived using the directional camera while the e-puck revolves to obtain the omni-directional picture.

We presented a series of experiments with 20 e-puck robots that validate the efficiency of the algorithm. The e-pucks were programmed to simulate a system of two groups of disks. The desired target pattern was an annular structure around a common point of attraction, where the robots in each annulus represent disks of identical size. The percentage of incorrectly-ordered pairs of disks from different groups decreased as the size ratio of disks in different groups was increased. This percentage was, on average, below 0.5% for size ratios from 3.0 to 5.0. Moreover, for these size ratios, all segregation errors observed were due to mechanical failures that caused robots to stop moving. Demonstration trials with 30 robots divided into three groups were also successful. To the best of our knowledge, this is the first example of segregation in a swarm of physical robots with no explicit communication and where such a high level of accuracy is achieved.

Fully Decentralized Segregation

Based on the segregation algorithm implemented for the e-puck robots, an improved algorithm has been developed to achieve the Brazil nut effect segregation. In the new algorithm, the center of attraction to the swarm is not a dedicated infrared light source. Instead, each of the robots is attracted by any nearby robots that have their LEDs activated. Each of the robots has two rules to activate its LEDs: (i) by probability, (ii) when it sees any robot with LEDs on. Once an activation occurred, new activation is prohibited for a duration. The emergent behavior of such a set of rules make the robots move on average towards the centroid of the swarm. Such bias in motion replaces the need for attraction to a global reference point (simulating gravitational pull) in the Brazil nut effect. Combined with random motion and heterogeneous repulsion, annular structures will be formed.

Compared to the original segregation algorithm based on the Brazil nut effect, the new algorithm does not contain a global center of attraction in a physical form. Therefore, the robots are not required to have consistent perception of the center.

In computer simulations, the algorithm successfully organized 61 robots into annular structures in 30 trials. A demonstration using 32 physical e-puck robots into two groups had also been achieved.

Moreover, the propagation of the LED signals had been studied more thoroughly using the data collected in the trials. From the direct observation, the robots near the center of the formation appeared to fire the LED signal more frequently than those near the edge of the formation. The data plot confirmed this observation; it revealed that a radial gradient exists in the firing rate of the LED signals within the formation.

A further cellular-space-based simulation had been constructed to study the signal propagation phenomenon observed in the segregation algorithm. Again, the radial gradient of the firing rate emerged across the board. In this study, the cells are static and only able to read information from four neighbor cells of the space, this means the radial gradient which emerges from the firing rules is independent of the segregation algorithm.

7.2 General Conclusions and Future Work

As shown in this thesis, several tasks have been achieved using a swarm of robots with no capability of explicit communication nor specific physical mechanisms such as grippers or specifically designed bodies. The swarm systems have good scalability; despite the fact that an optimal number of robots may exist, changes in the number of robots used may not affect the task performance dramatically. The robustness of the systems was also demonstrated through numerous physical experiments; regardless of the task considered and despite many failures in individual robots, the overall successful rate of the trials was always high.

Nevertheless, these systems have potential to be improved or expanded further.

For the cooperative transport strategy, the potential of transporting object of concave shapes could be investigated. Furthermore, improved individual behavior could be studied that allows multiple layers of robots to push objects that are heavy but small in surface area [50]. The possibility of controlling the orientation of the object during the transportation could also be studied. To expand the system, the goal could also be one of a series of way points formed by a group of robots (e.g. mimicking a trail of virtual pheromones [89][123]). Such a system may accomplish a more complex cooperative transport task autonomously. To fully expand the work to 3-D environments is also a valuable topic to study. For example, the conceptual robot used in the simulation could be developed.

In principle, the Brazil-nut-effect-based segregation could form annular structures in 3-D environments as well [39]. Future work could be aimed to implement a 3-D version of either of the two segregation controllers described in this thesis. Furthermore, the radial gradient phenomenon observed in the signaling based segregation controller can be utilized in collective localization algorithms. In current studies (e.g. [86][104]), global gradients typically need to be constructed based on a specified global center or 'root' while explicit communication among the robots is required. Our mechanism of gradient construction may be used to remove such assumptions despite relatively low hardware requirements.

Due to the simplicity of the controllers studied in this thesis, they are suitable for implementation on a large number of robots with limited capabilities. In the long term, swarms of such robots could be implemented at very small scales. Potential applications for swarms of such minimalist robots could be the delivery

of drugs through the vascular network of humans or forming functional structures at the microscopic level.

Appendix A

Appendices

A.1 Bullet Physics Model of e-puck

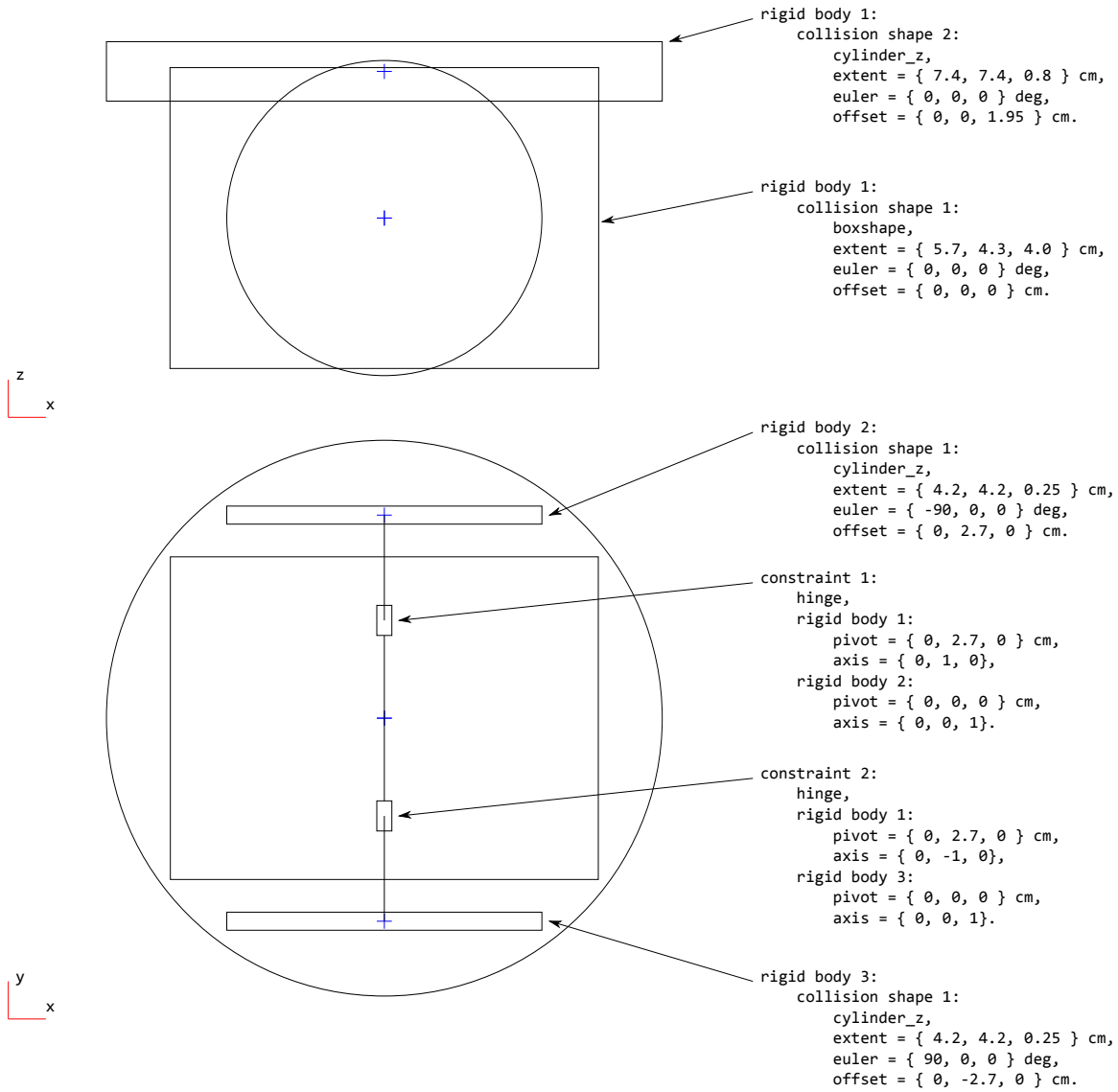


Figure A.1: This is the Bullet Physics model of the e-puck.

Bibliography

[1] M. Amos and O. Don. Swarm-based spatial sorting. *International Journal of Intelligent Computing and Cybernetics*, 1(3):454–473, 2008.

[2] T. Arai, E. Pagello, and L. E. Parker. Editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics*, 18(5):655–661, 2002.

[3] R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

[4] R. C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[5] R. C. Arkin and J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *7th International Workshop on Advanced Motion Control*, pages 455–461. IEEE, 2002.

[6] T. Balch and R. C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1:27–52, 1994. 10.1007/BF00735341.

[7] J. Bard and I. Lauder. How well does turing's theory of morphogenesis work? *Journal of Theoretical Biology*, 45(2):501–531, 1974.

[8] G. Barker and M. Grimson. The physics of muesli. *New Scientist*, 126(1718):37–40, 1990.

[9] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin. Massive uniform manipulation: Controlling large populations of simple robots with a common input signal. In *Proceedings of the 2013 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2013)*, pages 520–527. IEEE Computer Society Press, Los Alamitos, CA, 2013.

[10] R. A. Becker, J. M. Chambers, and A. R. Wilks. *The new S language. A programming environment for data analysis and graphics.* Chapman & Hall, London, 1988.

[11] R. Beckers, J. Deneubourg, S. Goss, and J. Pasteels. Collective decision making through food recruitment. *Insectes Sociaux*, 37(3):258–267, 1990.

[12] G. Beni. The concept of cellular robotic system. In *Proceedings of 1988 IEEE International Symposium on Intelligent Control*, pages 57–62. IEEE, 1988.

[13] G. Beni. From swarm intelligence to swarm robotics. In *Swarm Robotics*, pages 1–9. Springer, 2005.

[14] G. Beni and J. Wang. Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712. Springer, 1993.

[15] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. Pratt. Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proceedings of the IEEE*, 99(9):1470–1481, 2011.

[16] H. Bojinov, A. Casal, and T. Hogg. Emergent structures in modular self-reconfigurable robots. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*, volume 2, pages 1734–1741, 2000.

[17] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.

[18] N. Bowden, A. Terfort, J. Carbeck, and G. M. Whitesides. Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276(5310):233–235, 1997.

[19] S. Camazine, J.-L. Deneubourg, N. R.Franks, J. S. G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*, pages 217–284. Princeton Univ. Press, Princeton, NJ, 2001.

[20] S. Camazine, J.-L. Deneubourg, N. R.Franks, J. S. G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*, pages 189–215. Princeton Univ. Press, Princeton, NJ, 2001.

[21] S. Camazine and J. Sneyd. A model of collective nectar source selection by honey bees: Self-organization through simple rules. *Journal of Theoretical Biology*, 149(4):547–571, 1991.

[22] Y. U. Cao, A. S. Fukunaga, and A. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[23] G. Caprari and R. Siegwart. Mobile micro-robots ready to use: Alice. In *Proceedings of the 2005 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2005)*, pages 3295–3300. IEEE, 2005.

[24] J. Chen. e-puck embedded system library. `https://github.com/jianingchen/epucklib`, 2014.

[25] J. Chen. Online supplementary material. `http://naturalrobotics.group.shef.ac.uk/supp/thesis_jianingchen`, 2014.

[26] J. Chen, M. Gauci, and R. Groß. A strategy for transporting tall objects with a swarm of miniature mobile robots. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, pages 863–869. IEEE Computer Society Press, Los Alamitos, CA, 2013.

[27] P. Cheng, J. S. Pang, J. Fink, and V. Kumar. Cooperative towing with multiple robots. *Journal of Mechanisms and Robotics*, 1(1):011008, 2009.

[28] C. M. Cianci, X. Raemy, J. Pugh, and A. Martinoli. Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Swarm Robotics*, pages 103–115. Springer, 2007.

[29] E. Şahin, T. Labella, V. Trianni, J.-L. Deneubourg, P. Rasse, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, and M. Dorigo. Swarm-bot: Pattern formation in a swarm of self-assembling mobile robots. In *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, volume 4. IEEE Computer Society Press, Los Alamitos, CA, 2002.

[30] T. Czaczkes and F. Ratnieks. Simple rules result in the adaptive turning of food items to reduce drag during cooperative food transport in the ant pheidole oxyops. *Insectes Sociaux*, 58(1):91–96, 2011.

[31] T. J. Czaczkes and F. L. Ratnieks. Cooperative transport in ants (hymenoptera: Formicidae) and elsewhere. *Myrmecol. News*, 18:1–11, 2013.

[32] C. Detrain and J.-L. Deneubourg. Self-organized structures in a superorganism: do ants behave like molecules? *Physics of Life Reviews*, 3(3):162–187, 2006.

[33] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, et al. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2-3):223–245, 2004.

[34] M. Dorigo, E. Tuci, R. Groß, V. Trianni, T. H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi, et al. The swarm-bots project. In *Swarm Robotics*, pages 31–44. Springer, 2005.

[35] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for swarm robots. In *Proceedings of the 1993 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 1993)*, volume 1, pages 441–447. IEEE, 1993.

[36] J. Fink, M. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, pages 1471–1476. IEEE Computer Society Press, Los Alamitos, CA, 2008.

[37] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *International Journal of Robotics Research*, 30(3):324–334, 2011.

[38] D. Floreano, F. Mondada, D. Floreano, D. Floreano, F. Mondada, and F. Mondada. *Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot*. ETH-Zürich, 1994.

[39] S. Foster and R. Groß. Forming nested 3D structures based on the Brazil nut effect. In *Proceedings of the 12th Conference Towards Autonomous Robotic Systems (TAROS 2011)*, volume 6856 of *Lecture Notes in Artificial Intelligence*, pages 394–395, Berlin, Germany, 2011. Springer-Verlag.

[40] N. R. Franks and A. B. Sendova-Franks. Brood sorting by ants: distributing the workload over the work-surface. *Behavioral Ecology and Sociobiology*, 30(2):109–123, 1992.

[41] S. Garnier, J. Gautrais, and G. Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.

[42] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß. Self-organized aggregation without computation. *International Journal of Robotics Research*, 33(8):1145–1161, 2014.

[43] B. Gerkey and M. Mataric. Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[44] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12(1):30–39, 1972.

[45] R. L. Graham and N. J. Sloane. Penny-packing and two-dimensional codes. *Discrete & Computational Geometry*, 5(1):1–11, 1990.

[46] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130, 2006.

[47] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarm-bots. *IEEE Transactions on Robotics*, 22(6):1115–1130, 2006.

[48] R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 852–861. Springer, Berlin, Germany, 2004.

[49] R. Groß and M. Dorigo. Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9):1490–1508, 2008.

[50] R. Groß and M. Dorigo. Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, 1(1–2):1–13, 2009.

[51] R. Groß, S. Magnenat, and F. Mondada. Segregation in swarms of mobile robots based on the Brazil nut effect. In *Proceedings of the 2009 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2009)*, pages 4349–4356. IEEE Computer Society Press, Los Alamitos, CA, 2009.

[52] R. Groß, F. Mondada, and M. Dorigo. Transport of an object by six pre-attached robots interacting via physical links. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, pages 1317–1323. IEEE Computer Society Press, Los Alamitos, CA, 2006.

[53] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tâche, I. Said, V. Durier, S. Canonge, J. M. Amé, et al. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158, 2007.

[54] B. Hölldobler and E. O. Wilson. The multiple recruitment systems of the african weaver ant; (latreille) (hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, 3:19–60, 1978. 10.1007/BF00300045.

[55] Y. Hu, L. Wang, J. Liang, and T. Wang. Cooperative box-pushing with multiple autonomous robotic fish in underwater environment. *IET Control Theory and Applications*, 5(17):2015–2022, 2011.

[56] A. Huth and C. Wissel. The simulation of the movement of fish schools. *Journal of Theoretical Biology*, 156(3):365–385, 1992.

[57] Y. Ikemoto, Y. Hasegawa, T. Fukuda, and K. Matsuda. Gradual spatial pattern formation of homogeneous robot group. *Information Sciences*, 171(4):431–445, 2005.

[58] Y. Inoue, T. Tohge, and H. Iba. Cooperative transportation system for humanoid robots using simulation-based learning. *Applied Soft Computing*, 7(1):115–125, 2007.

[59] M. P. Jullien Remi. A mechanism for particle size segregation in three dimensions. *Nature*, 344(6265):425–427, 2005.

[60] A. J. Koch and H. Meinhardt. Biological pattern formation: from basic mechanisms to complex structures. *Reviews of Modern Physics*, 66:1481–1507, Oct 1994.

[61] S. Kondo and T. Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620, 2010.

[62] C. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101, 2000.

[63] C. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101, 2000.

[64] C. R. Kube and H. Zhang. Collective robotics: From social insects to robots. *Adaptive Behavior*, 2(2):189–218, 1993.

[65] C. R. Kube and H. Zhang. Task modelling in collective robotics. In *Robot Colonies*, pages 53–72. Springer, 1997.

[66] M. Kumar, D. Garg, and V. Kumar. Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3):743–748, 2010.

[67] V. Kumar and K. Waldron. Force distribution in closed kinematic chains. *IEEE Journal of Robotics and Automation*, 4(6):657–664, 1988.

[68] Y. Kume, Y. Hirata, Z. Wang, and K. Kosuge. Decentralized control of multiple mobile manipulators handling a single object in coordination. In *Proceedings of the 2002 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2002)*, volume 3, pages 2758–2763. IEEE Computer Society Press, Los Alamitos, CA, 2002.

[69] K. Lynch. The mechanics of fine manipulation by pushing. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation (ICRA 1992)*, volume 3, pages 2269–2276, May 1992.

[70] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6):533–556, 1996.

[71] L. Marsh and C. Onof. Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, 9(1):136–149, 2008.

[72] A. Martinoli, A. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29(1):51–63, 1999.

[73] M. J. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 432–441, 1993.

[74] M. J. Matarić, M. Nilsson, and K. T. Simsarian. Cooperative multi-robot box-pushing. In *Proceedings of the 1995 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 1995)*, volume 3, pages 556–561. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[75] H. Meinhardt. *The algorithmic beauty of sea shells*. Springer, 2009.

[76] H. Meinhardt and A. Gierer. Pattern formation by local self-activation and lateral inhibition. *Bioessays*, 22(8):753–760, Aug. 2000.

[77] H. Meinhardt and H. Meinhardt. *Models of biological pattern formation*, volume 6. Academic Press London, 1982.

[78] C. Melhuish, A. B. Sendova-Franks, S. Scholes, I. Horsfield, and F. Welsby. Ant-inspired sorting by robots: the importance of initial clustering. *Journal of the Royal Society Interface*, 3(7):235–242, 2006.

[79] M. B. Miller and B. L. Bassler. Quorum sensing in bacteria. *Annual Reviews in Microbiology*, 55(1):165–199, 2001.

[80] N. Miyata, J. Ota, Y. Aiyama, J. Sasaki, and T. Arai. Cooperative transport system with regrasping car-like mobile robots. In *Proceedings of the 1997 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 1997)*, volume 3, pages 1754–1761. IEEE Computer Society Press, Los Alamitos, CA, 1997.

[81] F. Mondada, M. Bonani, A. Guignard, S. Magnenat, C. Studer, and D. Floreano. Superlinear physical performances in a swarm-bot. In *Advances in Artificial Life*, pages 282–291. Springer Berlin Heidelberg, 2005.

[82] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco, 2009.

[83] F. Mondada, L. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneuborg, and M. Dorigo. The cooperation of swarm-bots: physical interactions in collective robotics. *IEEE Robotics Automation Magazine*, 12(2):21–28, June 2005.

[84] F. Mondada, A. Guignard, M. Bonani, D. Bar, M. Lauria, and D. Floreano. Swarm-bot: From concept to implementation. In *Proceedings of the 2003 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1626–1631. IEEE, 2003.

[85] J. R. Moyne and D. M. Tilbury. The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proceedings of the IEEE*, 95(1):29–47, 2007.

[86] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information Processing in Sensor Networks*, pages 333–348. Springer, 2003.

[87] A. M. T. Ngouabeu, S. Miyashita, R. M. Füchslin, K. Nakajima, M. Göldi, and R. Pfeifer. Self-organized segregation effect on self-assembling robots. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems (Artificial Life XII)*, pages 232–238. MIT Press, Cambridge, MA, 2010.

[88] S. Nolfi, J. Deneubourg, D. Floreano, L. Gambardella, F. Mondada, and M. Dorigo. Swarm-bots: swarm of mobile robots able to self-assemble and self-organize. *Ecrim News*, 53(1):25–26, 2003.

[89] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711, 2009.

[90] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[91] L. E. Parker. Current research in multirobot systems. *Artificial Life and Robotics*, 7(1-2):1–5, 2003.

[92] J. K. Parrish and L. Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.

[93] D. Payton, R. Estkowski, and M. Howard. Pheromone robotics and the logic of virtual pheromones. In E. Sahin and W. Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 45–57. Springer Berlin / Heidelberg, 2005.

[94] G. A. S. Pereira, M. F. M. Campos, and V. Kumar. Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 23(7–8):783–795, 2004.

[95] G. A. S. Pereira, V. Kumar, J. Spletzer, C. J. Taylor, and M. F. M. Campos. Cooperative transport of planar objects by multiple mobile robots using object closure. In *Experimental Robotics VIII*, pages 275–284. Springer Verlag, Berlin, Germany, 2002.

[96] P. Pipattanasomporn and A. Sudsang. Object caging under imperfect shape knowledge. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*, pages 2683–2688, May 2010.

[97] J. Riley, U. Greggers, A. Smith, D. Reynolds, and R. Menzel. The flight paths of honeybees recruited by the waggle dance. *Nature*, 435(7039):205–207, 2005.

[98] E. Rimon and A. Blake. Caging 2d bodies by 1-parameter two-fingered gripping systems. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA 1996)*, volume 2, pages 1458–1464, Apr 1996.

[99] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. 3-d relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1-2):5–20, 2012.

[100] W. L. Romey. Individual differences make a difference in the trajectories of simulated schools of fish. *Ecological Modelling*, 92(1):65–77, 1996.

[101] A. Rosato, K. J. Strandburg, F. Prinz, and R. H. Swendsen. Why the Brazil nuts are on top: size segregation of particulate matter by shaking. *Physical Review Letters*, 58(10):1038–1040, 1987.

[102] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA 2012)*, pages 3293–3298. IEEE, 2012.

[103] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal. Collective transport of complex objects by simple robots: theory and experiments. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2013)*, pages 47–54, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[104] M. Rubenstein, A. Cornejo, and R. Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[105] E. Sahin. Swarm robotics: From sources of inspiration to domains of application. In E. Sahin and W. Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Verlag, Berlin, Germany, 2005.

[106] T. C. Schelling. Models of segregation. *American Economic Review*, 59(2):488–493, 1969.

[107] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim. Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155, 2009.

[108] S. Sen, I. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *In Proceedings of the 12th National Conference on Artificial Intelligence*, pages 426–431, 1994.

[109] K. Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

[110] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM, 1994.

[111] M. Sitti. Micro-and nano-scale robotics. In *Proceedings of the 2004 American Control Conference*, volume 1, pages 1–8. IEEE, 2004.

[112] M. Sitti. Microscale and nanoscale robotics systems [grand challenges of robotics]. *Robotics & Automation Magazine, IEEE*, 14(1):53–60, 2007.

[113] P. Song and V. Kumar. A potential field based approach to multi-robot manipulation. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, volume 2, pages 1217–1222, 2002.

[114] W. Spears and D. Gordon. Using artificial physics to control agents. In *Proceedings of the 1999 International Conference on Information Intelligence and Systems*, pages 281–288, 1999.

[115] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski. Cooperative localization and control for multi-robot manipulation. In *Proceedings of the 2001 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2001)*, volume 2, pages 631–636. IEEE Computer Society Press, Los Alamitos, CA, 2001.

[116] D. Stilwell and J. Bay. Toward the development of a material transport system using swarms of ant-like robots. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation (ICRA 1993)*, volume 1, pages 766–771. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[117] J. H. Sudd. The transport of prey by an ant, pheidole crassinoda em. *Behaviour*, 16(3/4):295–308, 1960.

[118] A. Sudsang and J. Ponce. On grasping and manipulating polygonal objects with disc-shaped robots in the plane. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA 1998)*, volume 3, pages 2740–2746. IEEE Computer Society Press, Los Alamitos, CA, May 1998.

[119] A. Sudsang and J. Ponce. A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*, volume 2, pages 1068–1075, 2000.

[120] T. Sugar and V. Kumar. Decentralized control of cooperating mobile manipulators. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICRA 1998)*, volume 4, pages 2916–2921. IEEE Computer Society Press, Los Alamitos, CA, May 1998.

[121] T. Sugar and V. Kumar. Control of cooperating mobile manipulators. *IEEE Transactions on Robotics*, 18(1):94–103, 2002.

[122] K. Sugawara. An object manipulation by swarm robots inspired by granular convection phenomenon. In *IEEE/SICE 2013 International Symposium on System Integration (SII)*, pages 700–705, Dec 2013.

[123] K. Sugawara, T. Kazama, and T. Watanabe. Foraging behavior of interacting robots with virtual pheromone. In *Proceedings of the 2004 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 3074–3079. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[124] K. Sugihara and I. Suzuki. Distributed motion coordination of multiple mobile robots. In *Proceedings of the 5th IEEE International Symposium on Intelligent Control, 1990*, volume 1, pages 138–143, sep 1990.

[125] S. Suzuki, H. Asama, A. Uegaki, S. Kotosaka, T. Fujita, A. Matsumoto, H. Kaetsu, and I. Endo. An infra-red sensory system with local communication for cooperative multiple mobile robots. In *Proceedings of the 1995 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 1995)*, volume 1, pages 220–225. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[126] Y. Tan and Z.-y. Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.

[127] S. Thomas and C. Karl. A navigation algorithm for swarm robotics inspired by slime mold aggregation. In *Proceedings of the 2nd International Conference on Swarm Robotics*, pages 1–13, Berlin, Heidelberg, 2007. Springer-Verlag.

[128] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo. Evolving aggregation behaviors in a swarm of robots. In *Advances in Artificial Life*, pages 865–874. Springer, 2003.

[129] E. Tuci, R. Groß, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo. Cooperation through self-assembly in multi-robot systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(2):115–150, 2006.

[130] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.

[131] R. Vaughan. Massively multi-robot simulation in stage. *Swarm Intelligence*, 2(2-4):189–208, 2008.

[132] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron. Robot control of animal flocks. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*, pages 277–282. IEEE, 1998.

[133] R. T. Vaughan, K. Støy, G. S. Sukhatme, and M. J. Matarić. Blazing a trail: insect-inspired resource transportation by a robot team. In *Distributed Autonomous Robotic Systems 4*, pages 111–120. Springer, 2000.

[134] R. T. Vaughan, K. Stoy, G. S. Sukhatme, and M. J. Matarić. Go ahead, make my day: Robot conflict resolution by aggressive competition. In *In Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior*, pages 491–500. MIT Press, 2000.

[135] M. Waibel, L. Keller, and D. Floreano. Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–660, 2009.

[136] Y. Wang and C. W. de Silva. Cooperative transportation by multiple robots with machine learning. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006).*, pages 3050–3056. IEEE, 2006.

[137] Z. Wang, Y. Hirata, and K. Kosuge. Control multiple mobile robots for object caging and manipulation. In *Proceedings of the 2003 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1751–1756. IEEE Computer Society Press, Los Alamitos, CA, 2003.

[138] Z. Wang, Y. Hirata, and K. Kosuge. Control a rigid caging formation for cooperative object transportation by multiple mobile robots. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, volume 2, pages 1580–1585. IEEE Computer Society Press, Los Alamitos, CA, April 2004.

[139] Z. Wang, Y. Hirata, and K. Kosuge. Dynamic object closure by multiple mobile robots and random caging formation testing. In *Proceedings of the 2006 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 2006)*, pages 3675–3681. IEEE Computer Society Press, Los Alamitos, CA, Oct 2006.

[140] Z. Wang, Y. Hirata, Y. Takano, and K. Kosuge. From human to pushing leader robot: Leading a decentralized multirobot system for object handling. In *Proceedings of the 2004 International Conference on Robotics and Biomimetics (ROBIO 2004)*, pages 441–446, 2004.

[141] Z. Wang and V. Kumar. Object closure and manipulation by multiple cooperating mobile robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and*

Automation (ICRA 2002), volume 1, pages 394–399. IEEE Computer Society Press, Los Alamitos, CA, 2002.

[142] Z. Wang, E. Nakano, and T. Matsukawa. Realizing cooperative object manipulation using multiple behaviour-based robots. In *Proceedings of the 1996 IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS 1996)*, volume 1, pages 310–317, Nov 1996.

[143] Z. Wang, E. Nakano, and T. Takahashi. Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(5):537–549, Sept 2003.

[144] Z. Wang, M. Zhou, and N. Ansari. Ad-hoc robot wireless communication. In *Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 4045–4050, Oct 2003.

[145] R. Wehner. 'matched filters' - neural models of the external world. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 161:511–531, 1987. 10.1007/BF00603659.

[146] J. C. Williams and M. I. Khan. The mixing and segregation of particulate solids of different particle size. *The Chemical Engineer*, 269:19–25, 1973.

[147] M. Wilson, C. Melhuish, A. B. Sendova-Franks, and S. Scholes. Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17(2–3):115–136, 2004.

[148] S. Yamada and J. Saito. Adaptive action selection without explicit communication for multirobot box-pushing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 31(3):398–404, 2001.

[149] H. Yamaguchi, T. Arai, and G. Beni. A distributed control scheme for multiple robotic vehicles to make group formations. *Robotics and Autonomous Systems*, 36(4):125–147, 2001.

[150] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. Modular self-reconfigurable robot systems. *IEEE Robotics and Automation Magazine*, 14(1):43–52, 2007.

[151] M. Yim, Y. Zhang, and D. Duff. Modular robots. *IEEE Spectrum*, 39(2):30–34, 2002.

[152] V. Zykov, A. Chan, and H. Lipson. Molecubes: An open-source modular robotics kit. In *IROS 2007 Self-Reconfigurable Robotics Workshop*, 2007.

[153] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435(7038):163–164, 2005.